Universidade Federal do Rio de Janeiro

Núcleo de Computação Eletrônica

Sandro da Silva Reis

QUALIDADE DE SERVIÇO EM TELEFONIA IP: Parametrização de QoS para Serviços de Voz sobre IP

Rio de Janeiro

2008

Sandro da Silva Reis

QUALIDADE DE SERVIÇO EM TELEFONIA IP:

Parametrização de QoS para Serviços de Voz sobre IP

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro – NCE/UFRJ

Prof. Orientador Fabio David

M. Sc. – NCE-IM/UFRJ – Brasil

Rio de Janeiro

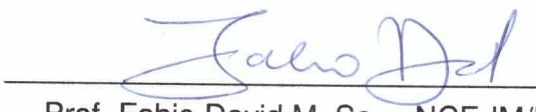
2008

Sandro da Silva Reis

QUALIDADE DE SERVIÇO EM TELEFONIA IP:
Parametrização de QoS para Serviços de Voz sobre IP

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro – NCE/UFRJ

Aprovada em outubro de 2008



Prof. Fabio David M. Sc. – NCE-IM/UFRJ – Brasil

AGRADECIMENTOS

Quero agradecer a Fátima Valéria, que muito me ajudou nessa caminhada e mostrou que com determinação e vontade somos capazes de fazer qualquer coisa e realizar qualquer sonho. Aos amigos de trabalho, Vinícios, Eli, Glaucio, Marcelo, Vitor e Emerson que me deram o apoio necessário para cumprir mais essa meta. Ao Frei Volney que disponibilizou tempo e recurso para minha formação, a minha esposa Daniela pelo apoio e carinho muito obrigado a todos.

RESUMO

REIS, Sandro da Silva. **QUALIDADE DE SERVIÇO EM TELEFONIA IP: Parametrização de QoS para Serviços de Voz sobre IP.** Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro.

Este trabalho procura investigar os efeitos causados por congestionamentos, perda de pacotes e jitter nas redes IP, prejudicando a funcionalidade do serviço de telefonia IP. Métodos de QoS (Qualidade de Serviço) são apresentadas como meios de melhorar o tráfego de voz sobre as redes IP e garantir uma boa qualidade no serviço. Mostrando através de parâmetros usados nas várias técnicas de QoS, que se pode chegar em um nível ótimo de qualidade, minimizando congestionamentos, perda de pacotes e jitter, para garantir um bom funcionamento.

ABSTRACT

REIS, Sandro da Silva. **QUALIDADE DE SERVIÇO EM TELEFONIA IP: Parametrização de QoS para Serviços de Voz sobre IP.** Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro.

This study seeks to investigate the effects caused by congestion, loss of packets and jitter in IP networks, damaging the functionality of the IP telephony service. Methods QoS (Quality of Service) are presented as ways to improve traffic from voice over IP networks and ensure a good quality in the services. Displaying through parameters used in the various techniques of QoS, which can have on a great level of quality, minimizing congestion, packet loss and jitter, to ensure a smooth operation.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

TDM	Time Division Multiplexing
PSTN	Public Switch Telephone Network
ISDN	Integrated Services Digital Network
TCP	Transfer Control Protocol
IP	Internet Protocol
UDP	User Datagram Protocol
ITU	International Telecommunications Union
RTP	Real-Time Transport Protocol
RTCP	Real-Time Control Transport Protocol
IETF	Internet Engineering Task Force
RSVP	Resource Reservation Protocol
IntServ	Integrated Services
DiffServ	Differentiated Services
QoS	Quality of Service
TOS	Type Of Service
DSCP	Differentiated Services Code Point
RED	Random Early Detection
WRED	Weighted Random Early Detection
SIP	Session Initiation Protocol
FRTS	Frame Relay Traffic Shapin
CAR	Commited Access Rate
GTS	Generic Traffic Shaping
CIR	Committed Information Rate
PHB	Per Hop Behaviors
CQ	Custom Queuing
PQ	Priority Queuing
FIFO	First In First Out
WFQ	Weighted Fair Queuing

SUMÁRIO

1 **INTRODUÇÃO**

Hoje com a criação de novos serviços e tecnologias usadas nas redes IP, como Voz sobre IP, há a necessidade de se ter um bom desempenho, mínima perda de pacotes, baixo jitter e controle de congestionamento. Baseado no estudo da Qualidade de Serviços em redes IP, este trabalho procura mostrar que aplicando parâmetros de QoS em redes IP é possível chegar a uma ótima utilização da voz sobre redes IP. O trabalho é baseado em dois modelos de classes de serviços para tráfego Internet que foram desenvolvidos pela IETF, conhecidos como IntServ e DiffServ, serviços integrados [RFC 1633] e serviços diferenciados [RFC 2475] respectivamente.

A telefonia analógica usa a modulação de sinais elétricos ao longo de um cabo para transportar voz. Apesar de ser uma tecnologia antiga, a transmissão analógica ainda tem muitas vantagens: ela é simples e mantém o atraso de fim a fim na transmissão de voz muito baixo, uma vez que o sinal se propaga ao longo do cabo quase na velocidade da luz. Também é barata quando há relativamente poucos usuários falando ao mesmo tempo e quando não estão muito distantes entre si. Mas a tecnologia analógica mais básica requer um par de cabos por conversação ativa, o que se torna impraticável e caro quando temos muitas conexões. Uma primeira melhoria da tecnologia analógica básica foi multiplexar várias conversações no mesmo cabo, usando uma freqüência de transporte separada para cada sinal. A menos que se use mesa de comutação manual, as chaves analógicas exigem uma grande quantidade de mecanismos eletromecânicos que são caros para comprar e manter.

Ruído parasítico é adicionado em todos os estágios da transmissão porque não há como dizer o que é sinal e o que é ruído, de maneira que o sinal não pode ser limpo.

Por todas essas razões, muitos países atualmente usam uma rede telefônica digital. Na maioria das vezes, a linha do assinante permanece analógica, mas o sinal analógico é convertido em um fluxo de sinais digitais na primeira central local. Normalmente, esse sinal tem uma taxa de transmissão de 64Kbps (uma amostra de 8 bits a cada 125μs).

Desta forma, vários canais de voz podem ser multiplexados ao longo da mesma linha de transmissão usando uma tecnologia chamada de multiplexação por

divisão no tempo *(Time Division Multiplexing - TDM*). Nessa tecnologia, o fluxo de dados digitais que representa uma única conversação é dividido em blocos (geralmente um octeto, também chamado de amostra) e blocos de várias conversações são intercalados em seqüência nos intervalos de tempo na linha de transmissão.

Com essa tecnologia digital, o ruído adicionado na estrutura de rede não influencia na qualidade da comunicação, uma vez que sinais digitais podem ser recuperados. Além disso, a multiplexão digital por divisão no tempo torna possível a comutação digital. O equipamento de comutação precisa apenas copiar o conteúdo de um intervalo de tempo da linha de transmissão de chegada em um outro intervalo de tempo da linha de transmissão de saída. Portanto, essa função de comutação pode ser executada por computadores. Entretanto, um pequeno atraso é introduzido por cada switch, porque para cada conversação em um intervalo de tempo torna-se disponível somente a cada T microssegundos e, em alguns casos, pode ser necessário esperar até T microssegundos para copiar o conteúdo de um intervalo de tempo em outro. Uma vez que T é igual a 125 microssegundos na maioria das redes digitais, isso geralmente é desprezível e o principal fator de atraso é simplesmente o tempo de propagação [TI02].

2 **TELEFONIA IP**

Na telefonia IP, as amostras de voz são acumuladas em pacotes IP e enviadas pela Internet. São necessários de 20 a 100 Kbps para uma chamada de voz, dependendo do codec utilizado. A figura 1 mostra o caminho sem terminais PSTN e ISDN.

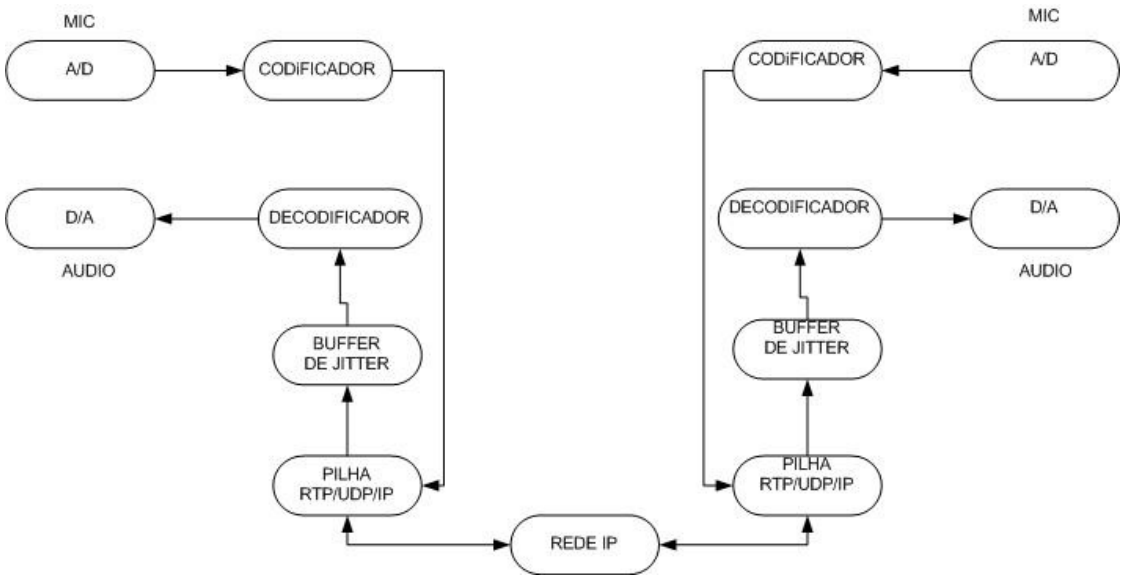


Figura 1 – Caminho de mídia sem terminais PSTN e ISDN.

2.1 ARQUITETURAS BÁSICAS

A seguir, são descritas as diversas arquiteturas utilizadas em ambientes de telefonia IP.

2.1.1 **Arquitetura PC a PC**

Nesta arquitetura, dois computadores providos com recursos multimídia, conectados à uma LAN ou através da RPT (Rede Pública de Telefonia) a um provedor de serviços Internet, se comunicam para troca de sinais de voz. Todo o tratamento do sinal de voz é feito nos computadores, sendo a chamada de voz estabelecida com base no endereço IP do receptor ou através de um nome que será convertido em endereço IP utilizando-se um serviço de diretório público. A figura 2 ilustra esse tipo de arquitetura. A arquitetura PC a PC possui uma variante onde o PC é substituído por um telefone com capacidade de codificação de voz e implementação do protocolo IP.

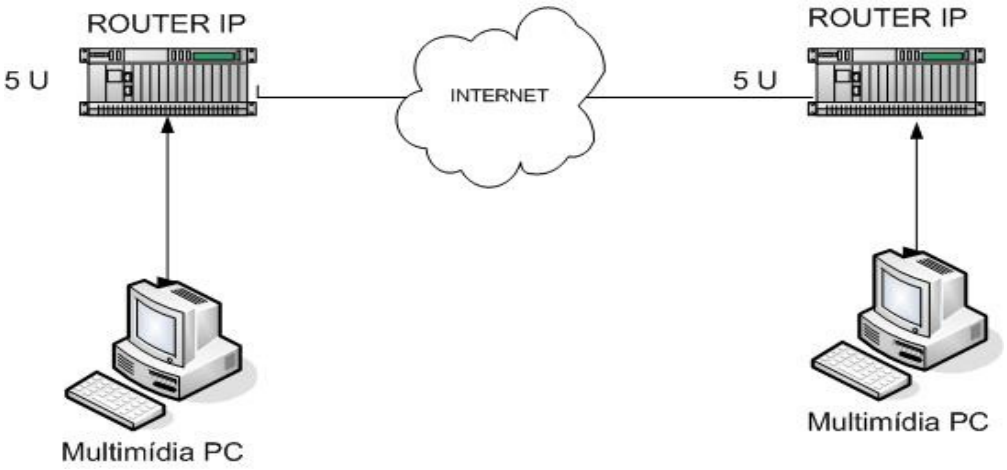


Figura 2 – Arquitetura PC a PC

2.1.2 **Arquitetura com Gateway**

Nesta arquitetura, um telefone padrão é utilizado para gerar e receber chamadas telefônicas através da Internet. O usuário chamador disca para o gateway de telefonia mais próximo de sua central telefônica, que reconhece e valida esse número telefônico do usuário chamador para fins de autenticação e bilhetagem e solicita a este o número do usuário de destino.

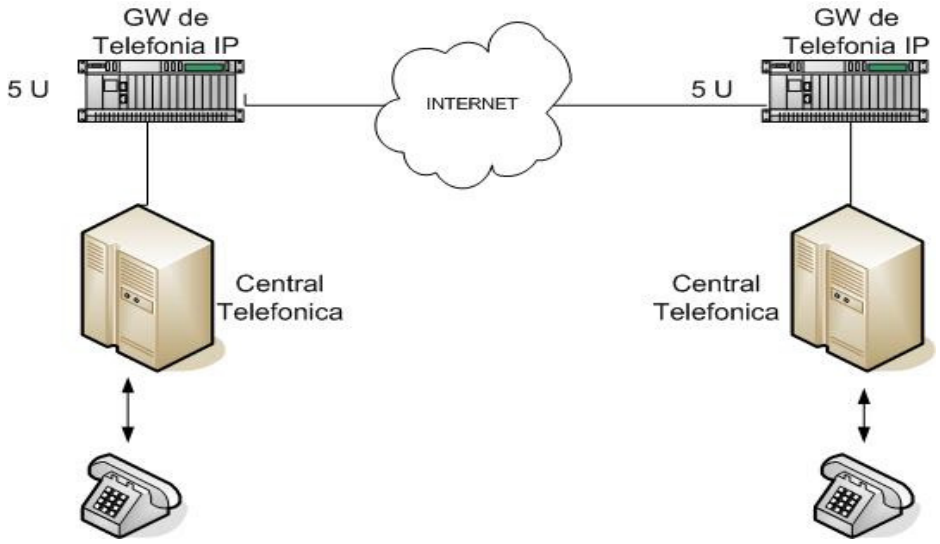


Figura 3 – Arquitetura com Gateway

O gateway de entrada identifica o gateway de saída mais próximo do usuário de destino e inicia com este uma sessão para transmissão de pacote de voz. O gateway de saída chama o telefone receptor e, após a chamada ser atendida, a comunicação fim a fim tem início, com o sinal de voz sendo enviado através de

datagramas IP entre os gateways. A codificação e o empacotamento do sinal de voz são feitos no gateway de origem, enquanto a decodificação e o desempacotamento são feitos no gateway de destino. A digitalização do sinal de voz pode ser feita na central, no Gateway ou mesmo no telefone (caso do RDSI, por exemplo).

2.1.3 **Arquiteturas Híbridas**

Naturalmente, esquemas híbridos das duas arquiteturas são possíveis e desejáveis. A figura 4 ilustra esse tipo de arquitetura. Nestas estruturas, um usuário de um telefone padrão origina ou recebe uma chamada para um usuário de PC ou telefone IP. Em tais situações deve existir um serviço de mapeamento ou translação de endereços IP em números telefônicos. Existem quatro caminhos unidirecionais neste caso: PC a PC, Gateway a Gateway, PC a Gateway, Gateway a PC. Em todas essas arquiteturas os pontos terminais (PC ou Gateways) devem empregar os mesmos esquemas de codificação de voz.

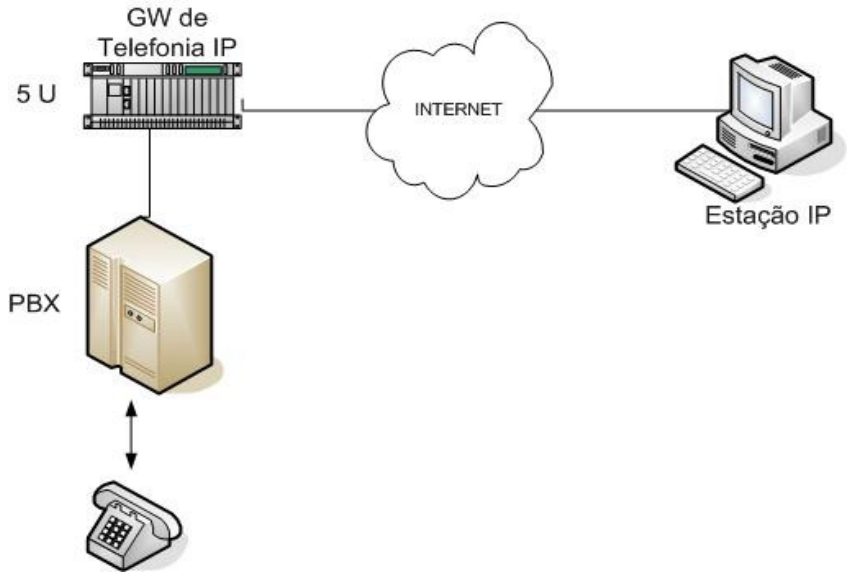


Figura 4 – Arquitetura Híbrida

2.2 O PROTOCOLO ITU H.323

O H.323 é uma recomendação do ITU para transmissão de voz e vídeo sobre redes TCP/IP, mais precisamente sobre aquelas redes que operem sobre Ethernet, Fast Ethernet, Gigabit Ethernet e Token Ring. Esse padrão foi aprovado em 1996 pelo grupo de estudos 16 do ITU. Devido à necessidade de um padrão para voz sobre IP, o H.323 foi revisado e surgiu a versão 2, adotada em janeiro de 1998. Na versão 3, foi adicionado suporte a comunicação gatekeeper-gatekeeper. A versão 4

teve como foco importantes áreas como, escalabilidade e confiabilidade. A versão 5 desse padrão possui poucas mudanças em relação a sua versão anterior, sendo a versão atual [DG04].

H.323 define padrões de voz e vídeo para uma infra-estrutura existente (redes TCP/IP) permitindo que os clientes possam usar aplicações sobre esses dados sem mudar a infra-estrutura de rede atual [DG04].

O padrão provê mecanismos de gerenciamento que permitem delimitar a quantidade de conferências simultâneas, bem como a quantidade de largura de banda destinada às aplicações H.323. Isso é necessário devido à característica de "consumidora de banda" das aplicações de tempo real. O H.323 também prevê mecanismos de contabilidade de uso dos recursos da rede, que pode ser usado para fins de cobrança [DG04].

A pilha de protocolos do padrão H.323 é apresentada na figura 5 de forma simplificada. Embora o RTP (*Real Time Protocol*) e o RTCP (*Real Time Control Protocol*) não façam parte do padrão H.323, esses protocolos são recomendados na implementação de aplicações seguindo esse padrão.

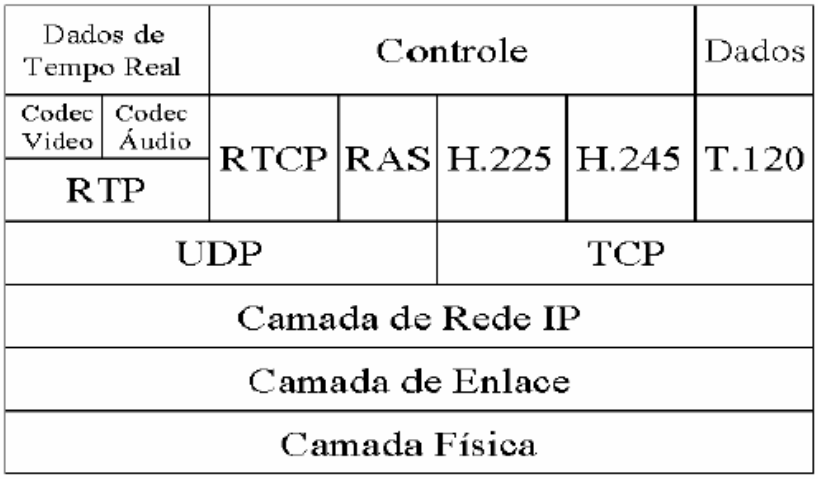


Figura 5 – Pilha de protocolos H.323

2.2.1 **O Procedimento de Inicialização de Chamadas H.323**

Para o estabelecimento de uma chamada H.323 fim a fim, são necessárias duas conexões TCP entre os dois terminais. Uma dessas conexões é destinada ao estabelecimento da chamada, enquanto a outra é responsável pelo controle da conferência H.323, bem como a troca de informações de capacidades entre os

terminais. Considerando a conexão TCP inicial, essa é estabelecida entre o terminal chamador e o terminal chamado. Nessa conexão, mensagens do tipo H.225.0 são trocadas entre esses terminais. Devido à semelhança desse padrão com o Q.931, costuma-se chamar esse canal de canal Q.931, ou, para fins mais didáticos, canal de sinalização de chamadas. Ao analisar esse princípio de comunicação, uma dúvida comum surge ao analisar a criação da conexão TCP inicial. Devido essa conexão ser iniciado a partir de um terminal, com um destino na outra extremidade da comunicação, o terminal chamador deve ter conhecimento não só do endereço de rede do destino, mas também da respectiva porta TCP.

De fato, a porta TCP para essa finalidade foi padronizada: a porta TCP 1720. Dessa forma, faz-se claro a necessidade de todos os terminais H.323 em "escutar" essa porta para tratar eventuais pedidos de abertura de conexão H.225.0.

A porta TCP 1720 foi padronizada para a comunicação inicial (*setup*). Contudo, nenhuma porta foi padronizada para a segunda comunicação TCP criada entre os terminais: o canal H.245. A porta TCP para estabelecimento desse canal é de caráter dinâmico.

Para que o terminal chamador saiba qual porta TCP usar para estabelecer o canal H.245 com o terminal chamado, existe um artifício na comunicação H.225.0 criada previamente. Durante essa troca inicial de mensagens, o terminal chamado indica ao chamador a porta TCP a ser utilizada na abertura do canal H.245.

2.2.2 **Encerrando uma chamada H.323**

O primeiro passo para terminar uma ligação é encerrar todos os canais lógicos H.245. Isso é feito através da mensagem H.245 *CloseLogicalChannel*. Após isso, o canal H.225.0 (Q.931), se ainda não tiver sido fechado (situação comum após a abertura do canal H.245), deve ser encerrado por um mensagem H.225.0 *ReleaseComplete*. Caso o terminal esteja conectado a um "gatekeeper", o próximo passo é encerrar a comunicação com ele através das mensagens DRQ (*Disengage Request*) e URQ (*Unregister Request*) do protocolo RAS (*Registration Admission Status*).

2.2.3 **Opções de chamada H.323**

A troca de mensagens necessárias antes do início efetivo da comunicação constitui-se em umas das grandes críticas a esse padrão. Tão prejudicial era

inicialmente essa característica que o padrão IETF SIP, introduzido após o H.323, possui um esquema de operação onde a comunicação efetiva ocorre logo “de início”. E isso se tornou uma das vantagens desse padrão em relação ao H.323.

Alguns esquemas adicionais foram desenvolvidos desde a primeira especificação do H.323 em 1996. Os mais importantes entre eles são o “*Fast Connect*” e o tunelamento H.245.

O procedimento “*Fast Connect*” permite que haja uma comunicação de áudio bidirecional após o recebimento da mensagem H.225.0 *Connect*. Para tanto, uma relação das capacidades (“codecs”) do terminal é enviada na mensagem SETUP. Dessa maneira, o *overhead* antes de ocorrer a comunicação efetiva é diminuído.

O tunelamento H.245 consiste no encapsulamento de mensagem desse padrão em mensagens *Setup* H.225.0. Dessa forma, o canal H.225.0 deve permanecer aberto durante toda a chamada. Existem inúmeras vantagens em se utilizar essa técnica, sendo de grande utilidade para terminais que estejam por trás de um *firewall*, pois apenas uma porta bem conhecida, a TCP 1720, será necessária.

2.3 PROTOCOLO DE INICIALIZAÇÃO DE SESSÃO – SIP

O protocolo SIP, definido pelo IETF, é um protocolo de sinalização para chamadas de telefonia sobre IP. Diferente do H.323, o protocolo SIP foi idealizado especificamente para a Internet. O SIP é um protocolo da camada de aplicação que permite a criação, modificação e estabelecimento de sessões com um ou mais participantes. O protocolo pode convidar participantes para iniciar uma sessão ou fazer parte de uma sessão já existente, que pode ter sido criada através de outros protocolos, como o H.323. O protocolo SIP suporta o mapeamento de nomes e serviços de re-direção. Estas facilidades capacitam o usuário a ter uma mobilidade entre terminais na rede [DGS03].

O protocolo SIP é um protocolo simples que possui as mesmas funcionalidades básicas que o protocolo H.323 possui, como a criação de um canal multimídia entre duas ou mais partes. Estas funcionalidades básicas são cinco:

- **Localização do Usuário:** É a localização do outro participante para o estabelecimento de uma sessão SIP;
- **Disponibilidade do Usuário:** É a determinação da disponibilidade do outro participante da sessão em se juntar na comunicação;

- **Capacidades do Usuário:** É a determinação dos parâmetros da mídia a ser trocada entre as duas ou mais partes envolvidas na sessão;
- **Estabelecimento/Finalização da Sessão:** É a capacidade de se estabelecer e finalizar uma sessão através da troca de mensagens SIP;
- **Gerenciamento da Sessão:** É a capacidade de gerenciar uma sessão estabelecida entre as partes envolvidas. Este gerenciamento é feito através da troca de mensagens entre as partes.

Além destas funcionalidades, o protocolo SIP pode ser usado com outros protocolos IETF para prover outros serviços multimídia. O protocolo SIP é usado, geralmente, com o protocolo SDP – *Session Description Protocol*, para descrever os parâmetros da mídia a ser trocada. Além do protocolo SDP, o SIP pode ser usado com o protocolo RTP – *Real-time Transport Protocol*, para transportar dados em tempo real, RTSP – *Real-time Streaming Protocol*, para controlar a entrega dos pacotes da mídia e o MEGACO – *Media Gateway Control Protocol*, para controlar *gateways*. Contudo, a operação SIP é independente destes protocolos.

O protocolo SIP sozinho não provê serviços. Ele provê primitivas que podem ser usadas para implementar diferentes tipos de serviços. Por exemplo, o SIP pode localizar um usuário e entregá-lo um objeto. Este objeto pode ser uma mensagem SDP para descrever um canal multimídia. Portanto, o protocolo SIP não trabalha sozinho para estabelecer um canal de comunicação multimídia.

2.3.1 **Mensagens SIP**

O Protocolo SIP usa mensagens textuais, codificadas em UTF-8 (*UCS Transformation Formats)*, para fornecer a sua sinalização. Estas mensagens são subdivididas em duas classes:

Mensagens de Requisição: Estas mensagens requisitam alguma ação do destinatário, e esta ação requisitada é chamada de método;

Mensagens de Resposta: As mensagens de resposta indicam o resultado do processamento da requisição, isto é, indicam a ação tomada pelo destinatário da requisição[DGS03].

As mensagens de requisição são distintas das mensagens de resposta pela linha inicial. A linha inicial das mensagens de requisição é chamada de linha de requisição. A linha de requisição tem o seguinte formato:

Método URI-Destino Versão-do-Protocolo.

Método: Atualmente existem seis métodos definidos na RFC 3261: **INVITE** e **ACK**, para estabelecer uma sessão; **BYE**, para finalizar uma sessão; **OPTIONS**, para obter as capacidades do servidor; **REGISTER**, para registrar o terminal SIP em um servidor de registros; e **CANCEL**, para cancelar uma tentativa de estabelecimento de sessão.

URI-Destino: Um URI – *Uniform Resource Identifier*, é um identificador de um terminal. Ele indica o usuário ou serviço para o qual esta requisição está sendo enviada. A forma geral de um URI em SIP é *username@domínio_sip*. Somente na mensagem REGISTER, o URI é o endereço IP do servidor de registros.

Versão-do-Protocolo: Ambas as requisições e respostas possuem a versão do protocolo na primeira linha da mensagem. A versão do protocolo mais recente é a 2.0. Portanto, o valor deste campo é SIP/2.0. Um exemplo da linha inicial de uma aquisição é:

INVITE sip:sandro@nce.ufrj.br SIP/2.0

A linha inicial, nas mensagens de resposta, é chamada linha de *status*. Isto se deve ao fato de que as respostas se referem a um *status* da requisição enviada para um servidor. A linha de *status* consiste de um código de status numérico, uma frase textual associada a este código, e a versão do protocolo SIP.

Versão-do-Protocolo Código-de-Status Frase

O código de *status* é um número de três dígitos que indica o resultado de uma tentativa de um servidor de entender e processar a requisição. A frase textual é uma descrição curta do código de *status*. O primeiro dígito do código de status define a classe da resposta. Os dois últimos dígitos não têm qualquer regra de categorização. A classificação das respostas é dada por:

1xx: Respostas Provisionais – Esta classe de resposta indica que o servidor recebeu a requisição e a está processando, mas ele não pode responder a requisição imediatamente. Portanto, esta resposta indica para o cliente que ele deve aguardar para que o servidor possa responder a requisição enviada.

2xx: Respostas de Sucesso – Esta classe de resposta indica que o servidor recebeu a requisição, processou e aceitou a requisição.

3xx: Respostas de Redireção – Esta classe indica que a requisição deve ser redirecionada para outro endereço, contido na resposta.

4xx: Respostas de Erro de Cliente – Esta classe de resposta indica que o servidor recebeu a requisição, mas o servidor não entendeu algum campo essencial da mensagem ou algum campo da mensagem não está bem formada.

5xx: Respostas de Erro no Servidor – Esta classe indica que houve alguma falha no processamento da resposta pelo servidor e ele não pode aceitar a requisição.

6xx: Respostas de Erro Global – Esta classe indica erros que podem ocorrer devido a algum requisito da mensagem que o servidor não suporte. Um exemplo da linha inicial de mensagem de resposta é:

SIP/2.0 200 Ok

2.3.2 **Entidades SIP**

Existem basicamente três entidades SIP: o terminal SIP, o proxy e o servidor de registros. O terminal SIP é aquele em que um usuário tem acesso aos serviços do protocolo SIP para estabelecer uma sessão. O terminal SIP pode ser um computador ou um telefone SIP.

O proxy é um elemento que pode ser classificado em: proxy com estado (*stateful*) e proxy sem estado (*stateless*). Um proxy com estado é aquele que quando recebe uma mensagem, se comporta como se ele mesmo estivesse mandando aquela mensagem. O proxy sem estado é simplesmente um encaminhador de mensagens. Ele somente verifica se a mensagem é para ele ou para o servidor de registros do domínio daquele proxy e reenvia a mensagem. Ele não se comporta como se fosse o requisitante.

O servidor de registros é a entidade SIP responsável por registrar os endereços SIP e os endereços reais dos usuários do protocolo SIP. Através do seu arquivo de registros, um proxy pode solicitar uma pesquisa para determinar se um determinado contato está registrado naquele servidor.

Todas as entidades SIP, com exceção do proxy sem estado, possuem todas as camadas do protocolo. O proxy sem estado possui somente a camada de transporte e um núcleo de processamento de mensagens do Proxy[DGS03].

2.4 O PROTOCOLO RTP E RTCP

O protocolo RTP é um produto do Grupo de Trabalho de Áudio e Vídeo do IETF definido pela RFC 3550 de 2003.

O RTP é um protocolo que fornece transporte fim-a-fim para aplicações que transmitem dados em tempo real. Ele não reserva recursos e não garante qualidade de serviços para serviços de tempo real, mas junto com RTCP fornece parâmetros para que uma aplicação possa compensar os efeitos do *jitter*. O RTP define um modo de formatar pacotes IP que carregam dados isócronos e inclui informação do tipo de dado transportado, timestamp, que é uma estampa de tempo que permite que um receptor controle a reprodução e número de seqüência, permitindo que o receptor detecte distribuição fora de ordem ou perda de pacote. O projeto do RTP/RTCP permite que esses protocolos sejam transportados acima de qualquer camada de protocolos de rede, mas seu principal uso é sobre o UDP, uma vez que o esquema de retransmissão do TCP não é adequado para dados que precisam ser transportados com uma latência muito baixa.

Uma parte importante do RTP é o suporte a tradução, capaz de alterar a codificação de um fluxo em uma estação intermediária. Outra caracteristica é o mixer, recebendo fluxo de dados de várias origens e combinando-os em um único fluxo e enviando o resultado. Quando várias pessoas em diversos locais diferentes participam de uma audioconferência usando IP, para minimizar o número de fluxos RTP, o grupo pode designar um mixer, determinando que cada local estabeleça uma sessão RTP com o mixer. O mixer combina os fluxos de áudio e envia o resultado como um único fluxo. A figura 6 demonstra o cenário de uma audioconferência , onde cada transmissor gera 64Kbps de tráfego de áudio, o mixer aceita cada stream e os combina em um único stream de 64Kbps[SC01].

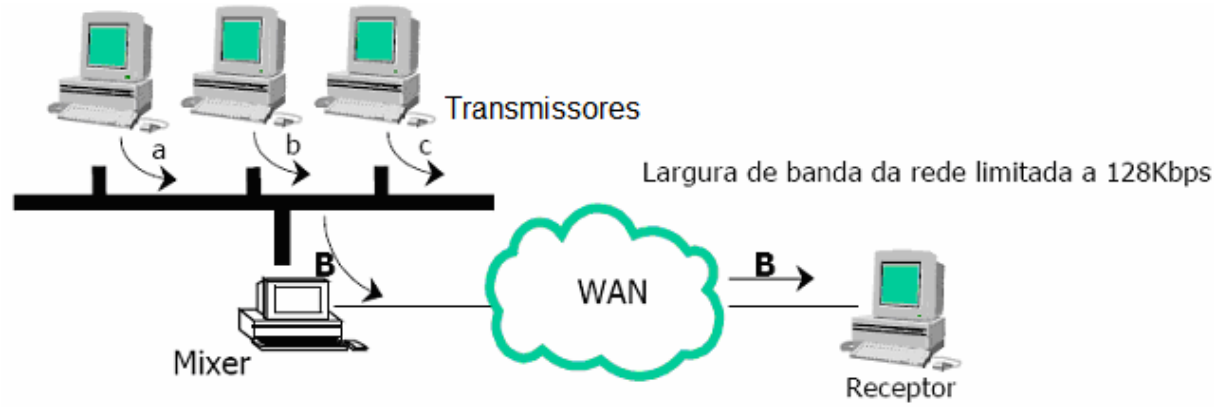


Figura 6 - Operação básica de um mixer

Os campos no cabeçalho RTP identificam a fonte e informam se houve combinação de fluxos. O campo indicado como SSRC (*Synchronization Source Identifier*) especifica a origem de um fluxo. Como cada origem precisa escolher um identificador de 32 bits único, o protocolo inclui um mecanismo para resolver conflitos, se houver. Quando um mixer combina vários fluxos, o mixer se torna a origem de sincronização para o novo fluxo. Porém, as informações sobre as origens iniciais não são perdidas, pois o mixer usa o campo CSRC (*Contributing Source Identifiers*) para fornecer os IDs de sincronização dos fluxos que foram mesclados. O campo CC fornece uma contagem das origens contribuinte, onde 15 origens no máximo podem ser listadas. A figura 7 mostra o cabeçalho do protocolo RTP.

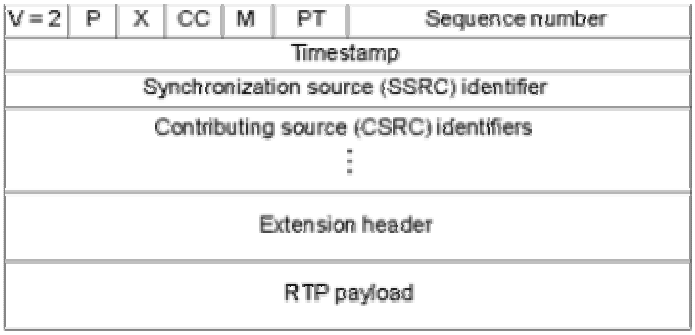


Figura 7 – Protocolo RTP

O RTP também opera com *multicasting* IP e com isso o uso de *mixers* se torna bastante atrativo para um ambiente *multicasting*, como por exemplo, em uma teleconferência onde há vários participantes. O *unicasting* exige que uma estação envie a cada participante uma cópia de cada pacote RTP de saída. Porém, com o *multicasting*, a estação só precisa enviar uma cópia do pacote, que será distribuída para todos os participantes. Além disso, se a mistura for usada, todas as origens podem difundir para um *mixer*, que os combina em um único fluxo antes do *multicasting*. Portanto, a combinação de *mixer* e *multicasting* resulta substancialmente menos datagramas sendo distribuídos a cada *host* participante.

Junto com o protocolo RTP, o RTCP é usado para transmitir aos participantes, de tempos em tempos, pacotes de controle relativos a uma sessão RTP em particular. Esses pacotes de controle podem incluir informações a respeito dos participantes e informações sobre o mapeamento dos participantes em suas fontes de fluxo individuais. Uma informação muito útil encontrada nos pacotes

RTCP é aquela que diz respeito à qualidade da transmissão na rede. As mensagens RTCP também são encapsuladas em UDP para transmissão e são enviadas usando o número de porta subsequente ao número de porta do fluxo RTP ao que elas pertecem. O RTCP define cinco diferentes tipos de mensagens [THO96]:

- *Sender Report* (SR): geradas pelos usuários que estão enviando as mídias (fontes RTP). Descrevem a quantidade de dados que está sendo enviada, ou seja, contadores cumulativos de pacotes e bytes enviados, e informações de *timestamp* que permitem a sincronização entre as diferentes mídias.
- *Receiver Report* (RR): geradas pelos participantes que estão recebendo as mídias informando sobre os níveis de qualidade na recepção do fluxo. O destino recebe, de tempos em tempos, uma descrição do tráfego gerado pelo fonte e contabiliza os dados recebidos. Um pacote RR carrega informações sobre a diferença entre o tráfego gerado e o tráfego recebido possibilitando o cálculo do impacto desses dados sobre a rede. Informações contidas nestes pacotes RTCP RR contém o maior número de seqüência recebido, número de pacote perdidos, atraso e variações de atraso entre pacotes. A fonte pode ou não alterar suas características de transmissão em função dos relatórios recebidos dos destinos.
- *Source Description* (SDES): são emitidas por fontes visando suprir mais informação sobre as mesmas. Exemplos incluem CNAME, um identificador único global semelhante ao formato dos endereço de e-mail. O CNAME é usado para resolver conflitos no valor SSRC (o transmissor original da mensagem) e associar diferentes fluxos de mídia gerados pelo mesmo usuário. Pacotes *Source Description* também identificam os participantes diretamente pelo nome, e-mail e número de telefone, localização geográfica do emissor, aplicação que esta gerando o fluxo e um texto adicional. Aplicações cliente podem mostrar o nome e informações de e-mail na interface do usuário permitindo aos participantes da sessão conhecerem outros participantes. Ele também permite aos participantes obterem informações de contato (como e-mail e telefone) para realizarem outras formas de comunicação (como iniciar uma sessão de conferência separado utilizando SIP).
- *Bye*: permite a uma fonte anunciar que está deixando de participar de

uma conferência.

- *Application Specific*: reservada para características específicas da aplicação. Projetada para as aplicações criarem novos tipos de mensagens.

Através desses pacotes com informações de controle, RTCP oferece os seguintes serviços:

- Monitoração de QoS e Controle de Congestionamento: Este é a função primária de RTCP. RTCP provê *feedback* para uma aplicação sobre a qualidade da recepção dos dados e contém informações necessárias para monitoração de QoS [BUS96]. As informações de controle são apropriadas para os enviadores dos dados RTP. Os recebedores podem determinar se um congestionamento está sendo local, regional ou global. Administradores de rede podem avaliar a performance da rede para distribuição *multicast*.
- Identificação do fonte: Em pacotes de dados RTP, fontes são identificadas por identificadores gerados aleatoriamente. Esses identificadores são diferentes para cada mídia particular gerada pelo fonte.
- Sincronização intermídia: Pacotes RTCP SR contém uma identificação de tempo real (NTP *Timestamp*) e um correspondente *timestamp* RTP (RTP *Timestamp*). Esses dois valores permitem sincronização de diferentes mídias, como por exemplo, sincronização labial de áudio e vídeo.
- Escalabilidade das informações de controle: Pacotes RTCP são enviados periodicamente entre participantes. Quando o número de participantes aumenta as informações de controle são balanceadas e o tráfego de controle é limitado.

2.5 CODIFICADORES DE VOZ

A maioria dos *codecs* de áudio utilizados para codificar voz utiliza uma freqüência de amostragem de 8.000 Hz e um filtro passa-faixa de 30 a 3.400 Hz, semelhante ao utilizado na telefonia convencional. Normalmente, esses *codecs* trabalham com pacotes de tamanho fixo com taxas de 20 a 50 pacotes por segundo e *bitrates* constante [RFC 1890]. Os *codecs* de áudio podem ser baseados em amostras ou *frames*. Em ambos os tipos de *codecs*, um certo número de amostra é

agrupado, podendo ser aplicado algum algoritmo de compressão ou não, gerando um pacote de áudio a ser transportado pelo RTP. O *timestamp* RTP do pacote de áudio gerado corresponde ao *timestamp* da primeira amostra do pacote [RFC 1889].

Nos *codecs* baseados em amostras, cada amostra é representada por um número inteiro de bits, sendo que esse número de bits pode ser variável, porém deve-se garantir um alinhamento de 8 bits para cada pacote de áudio gerado [RFC 1890]. Definindo-se o atraso desejado para o *codec* e a freqüência de amostragem, determina-se o número de amostras por pacote. Quanto maior for o número de amostras em um pacote, maior o atraso inserido. Dado um número de amostras por pacote, fica definido o número de pacotes por segundo. Dependendo do fator de compressão utilizado, o pacote será maior ou menor, isto é, o número de bits utilizados para representar cada amostra será maior ou menor[SME04].

Um *codec* hipotético baseado em amostras utilizando uma freqüência de amostragem de 8KHz, produz 40 pacotes por segundos e cada pacote transporta 200 amostras. Sem compressão, essas 200 amostras ocupariam 200 bytes. Nesse caso, o *bitrate* do *codec* seria de 64 Kbps. Se o *codec* possuir um fator de compressão de 10 vezes, cada amostra seria representada em media, por 0,8 bits, o *codec* passaria a ser de 6,4 Kbps e cada pacote carregaria apenas 20 bytes de dados. Considerando-se 12 bytes de cabeçalho RTP, 8 bytes de cabeçalho UDP, 20 bytes de cabeçalho IP e 18 bytes ocupados na camada de enlace (em redes ethernet), cada pacote de áudio comprimido ocuparia 78 bytes, resultando em um *bitrate* do canal de 24,960 Kbps, um valor 3,9 vezes maior que o *bitrate* do *codec* [SME04].

Nos *codecs* baseados em *frames*, um bloco de tamanho fixo é comprimido geralmente em um bloco também de tamanho fixo. Um ou mais *frames* podem ser encapsulados em um mesmo pacote RTP, mas quanto maior o número de *frames* em um pacote RTP, maior o atraso inserido, porque mais tempo precisa ser esperado antes de se gerar e transmitir o pacote [RFC 1890].

Um *codec* hipotético semelhante ao exemplo baseado em amostras, trabalhando com *frames* de 20 ms e produzindo blocos de 20 bytes, transmitindo um bloco desses por pacote, apresentaria o mesmo comportamento do exemplo anterior. Se ele transmitisse um número N de blocos por pacote, o pacote seria maior e o *overhead* diminuiria, mas o atraso entre os pacotes aumentaria N vezes. Pode-se calcular o *bitrate* de um *codec* baseado em frames de forma análoga ao

cálculo do *bitrate* de um *codec* baseado em amostras, necessitando-se apenas do tamanho do pacote e da taxa de pacotes por segundo[SME04].

A qualidade subjetiva de um *codec* de áudio pode ser definida pela fidelidade com relação ao áudio original, pela inteligibilidade do áudio codificado e pela facilidade de reconhecimento e identificação do locutor [SME04].

Uma forma padronizada de avaliação da qualidade subjetiva do áudio codificado foi concretizada na recomendação P.800 – *Methods for Subjective Determinetion of transmission Quality,* que emprega pontuação média por opinião (MOS – *Mean Opinion Score*). Amostras de voz pré-selecionadas são reproduzidas para homens e mulheres sob condições controladas, que atribuem notas de 1 a 5 para cada critério estabelecido. Um *codec* de voz com MOS igual a 4 possui uma qualidade subjetiva comparável a de uma chamada de longa distância em telefonia convencional. Os critérios adotados para se dar nota a qualidade de voz são apresentadas na tabela 1.

Tabela 1 – Avaliação de áudio através dos critérios de MOS – Fonte [SME04]

Pontuação	**Tipo de teste**			
	Escala de opinião para teste de conversação (MOS_C)	**Escala de opinião para teste auditivo (MOS)**	**Escala de esforço para a audição (MOS_{LE})**	**Escala de preferência de volume (MOS_{LP})**
5	Excelente	Excelente	Relaxamento completo possível, não requer esforço.	Muito mais alto que o preferível.
4	Bom	Bom	É necessário prestar atenção, mas não precisa de um esforço considerável para a compreensão do significado.	Mais alto que o preferível.
3	Razoável	Razoável	É necessário um esforço moderado.	Preferível
2	Pobre	Pobre	É necessário um esforço moderado.	Mais baixo que o preferível.
1	Ruim	Ruim	Não é possível compreender o significado.	Muito mais baixo que o preferível.

Através de modelos matemáticos, dos aparelhos auditivos e fonador humano foram criados métodos mais objetivos e reprodutíveis para se avaliar a qualidade subjetiva do áudio codificado, de forma a substituir a recomendação P.800.

A escolha do *codec* de áudio é realizada de acordo com o cenário de utilização do mesmo e os principais critérios considerados são: a banda ocupada pelo *codec*, seu atraso e sua qualidade subjetiva. Cada cenário oferece bandas e atrasos diferentes.

Dois *codecs* podem ter *bitrate* iguais, mas um possuir uma qualidade subjetiva melhor que o outro. Em um cenário onde banda não é um fator crítico, o *codec* de melhor qualidade subjetiva deve ser escolhido. Se ambos possuírem qualidades subjetivas próximas, pode-se escolher o *codec* de menor atraso. A tabela 2 compara alguns codecs de voz utilizados.

O PCM 8KHz mono não utiliza nenhum mecanismo de compressão, ele apenas quantiza e codifica cada amostra. Quantidade de bits utilizada na codificação pode ser 6, 7 ou 8, resultando em uma taxa de transmissão de 48, 56 e 64Kbps respectivamente. Sendo assim a perdas no PCM são devidas a quantização, à baixa freqüência de amostragem e o uso de filtros.

Tabela 2 – Tabela de comparação entre codecs

Codec	Algoritmo (amostra/*frame*)	Bit rate (kbps)	Atraso algorítmico (ms)	MOS	Complexidade (MIPS)
G.711	PCM (amostra)	48, 56 e **64**	0,125 por amostra	4,3	0,01
G.722	SB-ADPCM (amostra)	48, 56 e **64**	< 2	4,1	5
G.723.1	MP-MLQ/ACELP (*frame*)	5,3 e 6,4	30	3,7 e 4,0	11
G.726	ADPCM (amostra)	16, 24, **32**, 40	0,125 por amostra	4,1	2
G.727	EADPCM (amostra)	16, 24, **32**, 40	0,125 por amostra	4,1	2
G.728	LD-CELP (*frame*)	16	2,5	3,4	30
G.729	CS-ACELP (*frame*)	8	10	4,0	20
G.729 anexo A	CS-ACELP (*frame*)	8	15	3,8	11
GSM[8] 06.10	RPE-LTP (*frame*)	13,2	20	3,5 a 3,9	6

O atraso algorítmico dos *codecs* de áudio baseados em amostras e em frames dependem de quantas amostras ou frames há em cada pacote de áudio, respectivamente. Os *codecs* de voz possuem baixo *bitrate*, de 2,4 kbps a 64 Kbps, mas o *overhead* introduzido pelos protocolos RTP, UDP, IP e pela camada de enlace elevam muito o *bitrate* do canal.

3 **QUALIDADE DE SERVIÇO PARA TELEFONIA IP**

Para se ter uma boa qualidade em uma chamada de voz sobre uma rede IP, deve-se garantir a confiabilidade, reduzir ao mínimo o descarte de pacotes de voz (onde a perda e descartes no buffer de compensação de jitter não deve ultrapassar 1% para chamadas de alta qualidade), e reduzir o atraso total e o *jitter*. O pacote de voz precisa ter o atraso menor e o valor máximo ao concorrer com outros tipos de tráfego na rede, não devendo ser superior a 150ms para chamadas de alta qualidade, embora na prática atrasos de até 200ms são toleráveis. Deve-se ter também a mínima variação no atraso dos pacotes de voz (onde a variação máxima tolerável é entre 20 e 50 ms) que também depende do tamanho dos buffers adaptativos de compensação de jitter e dos outros atrasos que compõe o atraso fim-a-fim.

A Internet funciona atualmente na filosofia de melhor esforço, isto é, as mensagens que são enviadas para Internet são divididas em pacotes IP que trafegam e são encaminhados para seu destino de uma maneira mais rápida e confiável possível. Como o protocolo IP não é orientado a conexão, os pacotes podem chegar fora de ordem, visto que a decisão de qual caminho tomar é, geralmente, dinâmica e baseada na densidade de tráfego de cada enlace, mensurado em tempo real. Dois modelos de classes de serviços para tráfego Internet foram desenvolvidos pela IETF para adicionar recursos de qualidade de serviços necessários a pilha de protocolo TCP/IP. O primeiro modelo refere-se aos serviços integrados [RFC 1633], conhecido como IntServ, que fornece uma garantia absoluta na alocação dos recursos da rede. O segundo modelo refere-se aos serviços diferenciados [RFC 2475], conhecido como DiffServ, que provê um tratamento diferenciado a determinados tipos de fluxos.

3.1 O PADRÃO INTSERV

O modelo de Serviços Integrados é baseado na reserva de recurso, ou seja, antes que os dados sejam transmitidos, as aplicações primeiro devem configurar caminhos e reservar recursos, onde o protocolo RSVP é responsável por essa sinalização. O RSVP não é um protocolo de roteamento, e sim trabalhando em conjunto com este. É usado por uma aplicação para requisitar uma qualidade de serviço específica da rede. O protocolo atua tanto em máquinas do usuário quanto

em roteadores, responsabilizando-se, nesse caso, a estabelecer e manter as condições para o serviço requisitado. O RSVP negocia a reserva de recursos em um único sentido de cada vez, ou seja, de forma *simplex*. Com isso, ele trata distintamente receptores e transmissores, operando juntamente com a camada de transporte.

O RSVP oferece dois tipos de serviço:

- **Serviço de carga controlada**: uma aplicação que necessite de um serviço de carga controlada para um fluxo com determinadas características espera que a rede se comporte como se estivesse pouco carregada para aquele fluxo. A sessão pode assumir que uma percentagem muito alta de seus pacotes passará com sucesso através do roteador sem ser cortada e com atraso de enfileiramento muito próximo a zero. O serviço de carga controlada não fornece garantias quantitativas acerca do desempenho, ele não especifica o que constitui uma percentagem muita alta de pacotes, nem que qualidade de serviço será fornecida por um elemento de rede não sobrecarregado.
- **Serviço garantido**: esse tipo de serviço não só pede uma largura de banda especifica, mas também um atraso de tráfego máximo. Basicamente, uma sessão requisitando serviço garantido está requerendo que os bits em seus pacotes tenham uma taxa de transmissão garantida. Para este tipo de serviço todos os nós intermediários devem implementar os serviços garantidos.

No modelo IntServ, os roteadores devem ser capazes de reservar recursos a fim de fornecer QoS especial para fluxo de pacotes específicos do usuário. Neste caso, o estado específico dos fluxos deve ser mantido pelos roteadores. Três componentes implementam este modelo de protocolo de sinalização: rotina de controle de admissão, classificador de pacotes e escalonamento de pacotes.

3.1.1 **Controle de admissão**

É simples a idéia que está por trás do controle de admissão: quando algum novo fluxo deseja receber determinado nível de serviço, o controle de admissão examina os campos TSpec e RSpec de especificação do fluxo e tenta decidir se o serviço desejado pode ser fornecido a essa quantidade de tráfego, dados os

recursos atualmente disponíveis, sem permitir que qualquer fluxo devidamente autorizado receba pior serviço do que havia solicitado. Se ele puder oferecer esse serviço, o fluxo será admitido, se não, ele será negado [PD03].

O controle de admissão depende muito do tipo de serviço solicitado e da disciplina de enfileiramento empregada nos roteadores. Para um serviço garantido, precisamos ter um bom algoritmo para tomar uma decisão sim/não definitiva. A decisão é bastante simples se o enfileiramento imparcial ponderado for usado em cada roteador. Para um serviço de carga controlada, a decisão pode ser baseada na heurística, como por exemplo em uma última vez em que foi permitido que um fluxo com essa TSpec entrasse nessa classe, os roteadores na classe excederam o limite aceitável, e por isso é melhor negar ou os retardos atuais estão tão dentro dos limites que se deve ser capaz de aceitar um novo fluxo sem dificuldade [PD03].

O controle de admissão não deve ser confundido com política. O controle de admissão é uma decisão para cada fluxo, admitindo um novo fluxo ou não. Política é uma função aplicada com base em cada pacote, para certificar-se de que um fluxo esteja de acordo com a TSpec que foi usada para fazer a reserva. Se um fluxo não estiver de acordo com sua TSpec, pois está transmitindo em uma taxa superior a que havia sido especificada, então ele provavelmente interferirá com o serviço fornecido a outros fluxos e alguma ação corretiva precisará ser tomada. Existem várias opções, sendo a mais óbvia descartar os pacotes problemáticos. Entretanto, uma outra opção seria verificar se os pacotes realmente estão interferindo com os serviços de outros fluxos. Se não estiverem interferindo, os pacotes poderiam ser enviados depois de serem marcados com uma etiqueta “pacote não conforme”, pois havendo necessidade de descarte esses pacotes serão descartados primeiro [PD03].

3.1.2 **Classificador de pacotes**

A classificação de pacotes é feita examinando-se até cinco campos no pacote: o endereço de origem, o endereço de destino, o número do protocolo, porta de origem e porta de destino. Com base nessa informação, o pacote pode ser colocado na classe apropriada. Como exemplo, ele pode ser classificado na classe de carga controlada ou pode fazer parte de um fluxo garantido que precisa ser tratado separadamente de todos os outros fluxos garantidos. Existe um mapeamento entre a informação específica do fluxo no cabeçalho do pacote um único

identificador de classe, que determina como o pacote será tratado na fila. Para fluxos garantidos, isso pode ser um mapeamento um para um, enquanto para outros serviços, poderia ser muitos para um [PD03].

3.1.3 **Escalonamento de pacotes**

Após a classificação, o escalonador seleciona para a transmissão o pacote de modo a satisfazer os requisitos de QoS. O escalonador de pacotes gerencia a retransmissão dos diferentes pacotes usando um conjunto de filas e outros mecanismos tais como temporizadores.

A função básica do escalonamento é implementar uma política para servir os pacotes na fila de saída e escolher que pacote enviar em um dado momento, onde o mecanismo para selecionar entre um conjunto de pacotes é conhecido como *scheduler* ou escalonador de tráfego. Existem alguns aspectos a serem considerados ao construir um *scheduler*.

Justiça – O *scheduler* deve garantir que os recursos consumidos por um fluxo estejam dentro da quantidade atribuída para o fluxo.

Retardo – Os pacotes em um determinado fluxo não devem ser retardados excessivamente.

Adaptabilidade – Se um determinado fluxo não tiver pacotes para enviar, o *scheduler* deve dividir a largura de banda extra entre outros fluxos proporcionalmente aos seus recursos atribuídos.

Veremos abaixo alguns algoritmos para escalonamento de pacotes.

3.1.3.1 FIFO – First In First Out (Atendimento por ordem de chegada)

Também conhecido como *first-come first-served* (FCFS – Primeiro a chegar, primeiro a ser servido), o FIFO não traz o conceito de classes de tráfego. Tudo o que ele faz é enviar os pacotes para a interface de saída na ordem de chegada, ou seja, o primeiro que chega é o primeiro a sair. O FIFO é o método mais rápido de enfileiramento e pode ser o mais efetivo para enlaces de banda larga com pequeno retardo e congestionamento mínimo[JC05].

3.1.3.2 PQ – Priority Queuing (Fila com prioridade)

Este mecanismo classifica os pacotes conforme um conjunto de regras pré-definidas e envia o pacote para a fila correspondente. O *Priority Queuing* (PQ) dá

prioridade absoluta para as filas de maior prioridade, ou seja, uma fila de menor prioridade somente envia tráfego após as filas de maior prioridade estarem vazias. A figura 8 ilustra o mecanismo de prioridade.

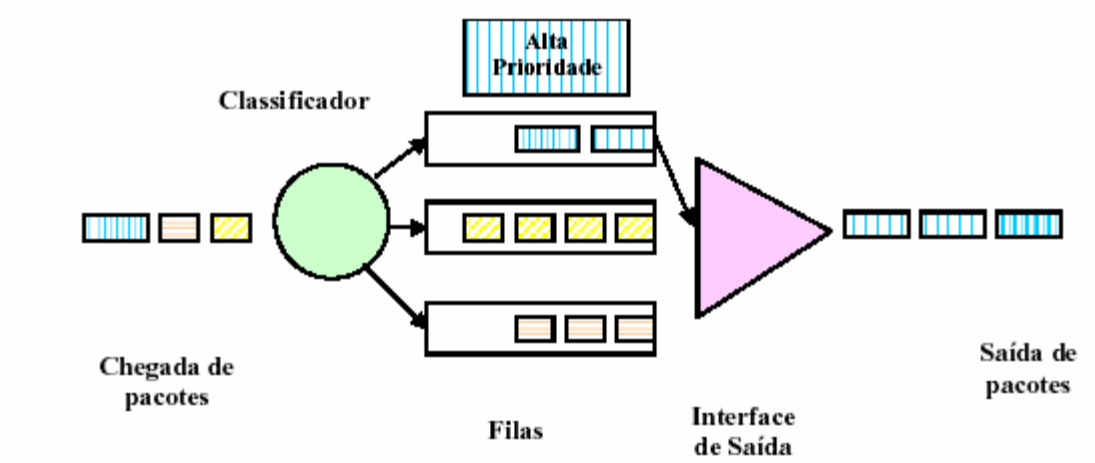


Figura 8 – Processo de priorização de fila

O tamanho máximo de uma fila é limitado: quando a fila excede o limite de tamanho definido, todos os pacotes adicionais são descartados. Esse mecanismo possui a vantagem de garantir a qualidade de serviço para o tráfego da fila de maior prioridade; em contrapartida, os tráfegos de menor prioridade podem ficar totalmente bloqueados [JC05].

3.1.3.3 CQ – Custom Queuing (Filas sob medida)

O *Custom Queuing* (CQ) permite ao administrador do sistema configurar o número de filas, o limite do tamanho da fila e a porcentagem da banda disponível para cada fila quando esta é servida. O CQ permite ao administrador dividir os recursos da rede entre diversas aplicações com largura de banda e retardo mínimos específicos.

De forma a garantir a divisão da banda entre as filas, o CQ especifica o número de pacotes que devem ser servidos para cada classe de tráfego. As filas são servidas periodicamente através do método *roundrobin*, ou seja, o roteador permite cada fila enviar uma certa quantidade de *bytes* antes de mudar para a próxima. Se uma fila está vazia então o roteador passa para a próxima que contém pacotes para enviar.

O roteador só pode enviar pacotes inteiros, sendo assim é necessário determinar quantos pacotes correspondem e quantidade de *bytes* reservados para transmissão em uma fila. Entretanto, o tamanho dos pacotes dos protocolos não são iguais. Assim, torna-se necessário a elaboração de um mecanismo um pouco mais complexo para determinar quantos pacotes cada fila transmitirá durante o seu ciclo. O procedimento será o seguinte:

Primeiro, determina-se para cada fila, a porcentagem da banda alocada pelo tamanho do pacote. Por exemplo, sejam 3 filas e as porcentagens alocadas iguais a 50% para a fila A, 30% para a B e 20% para a C e o tamanho dos pacotes conforme mostrado na figura 9.

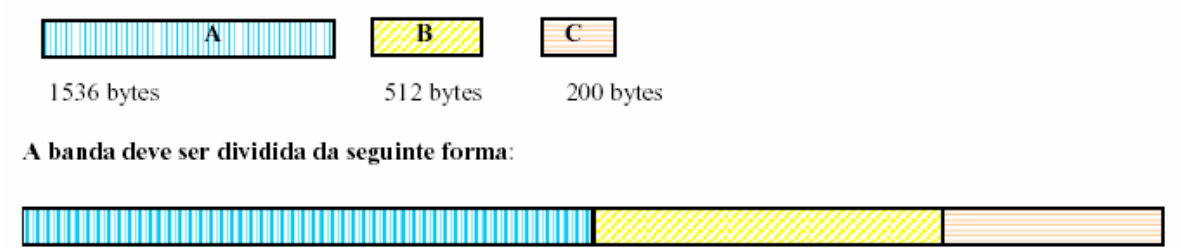


Figura 9 – Divisão da largura de banda

Agora se calcula quantos pacotes A podem ser colocados dentro da largura de banda reservada para A, quantos pacotes do protocolo B e o número de pacotes do protocolo C. Supondo que a largura de banda seja igual a 512kBytes/s ou 524.288 Bytes/s, tem-se:

Para o protocolo A: (524.288 x 50%) / 1536 = 171 pacotes/s

Para o protocolo B: (524.288 x 30%) / 512 = 307 pacotes/s

Para o protocolo C: (524.288 x 20%) / 200 = 524 pacotes/s

Finalmente, divide-se o número de pacotes encontrados para cada um dos protocolos pelo menor valor encontrado, ou algum outro mais conveniente. Esse procedimento tem por objetivo minimizar o retardo, impedindo que uma fila tenha que esperar a outra transmitir centenas de pacotes de uma vez. Em cada ciclo tem-se que:

A Fila A transmite 1 pacote (171 / 171)

A Fila B transmite 2 pacotes (307 / 171)

A Fila C transmite 3 pacotes (524 / 171)

Observe que apesar da fila A transmitir apenas 1 pacote por vez, conforme é ilustrado na figura 10, o tamanho dos pacotes são maiores que os outros, sendo assim ela transmitirá 1536 bytes por ciclo, a fila B transmitirá 1024 bytes e a fila C transmitirá 600 bytes por ciclo. Calculando a porcentagem da banda reservada verifica-se que A utiliza 48,6%, B utiliza 32,4% e C utiliza 19%, valores muito próximos aos desejados.

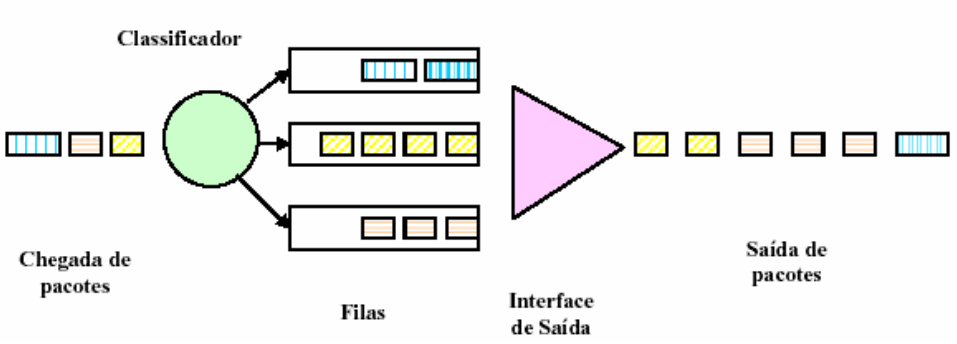


Figura 10 – Utilização de Custom Queuing

O CQ oferece a garantia da banda passante para um tráfego específico em um determinado ponto de congestionamento, entretanto a sua configuração é estática, ou seja, ele não se adapta às mudanças das condições da rede ou mudanças de tráfego[JC05].

3.1.3.4 WFQ – Weighted Fair Queuing (Enfileiramento Imparcial Ponderado)

O WFQ é um método de escalonamento automático que oferece uma alocação justa de banda para todo o tráfego da rede. O WFQ utiliza pesos para identificar e classificar os tráfegos e determinar quanto de banda um tráfego possui em relação a outros tipos de tráfegos.

Desta forma, o WFQ resolve as limitações do mecanismo FIFO. Impede que uma rajada de um tipo de tráfego monopolize toda a banda disponível, pois igualmente ao CQ (*Custom Queuing*), reserva um pequeno período de tempo, proporcional ao peso atribuído aquele tipo de tráfego, para cada fila transmitir seus pacotes.

O WFQ possui a vantagem de não bloquear totalmente um tráfego conforme pode acontecer com o *Priority Queuing*, pois todos os tipos de tráfego possuem um mínimo de banda garantido. Além disso, o WFQ não é configurado estaticamente

como o CQ, automaticamente se adapta às mudanças das condições do tráfego da rede.

O WFQ utiliza o campo *IP Precedence* (3 bits) do cabeçalho IP para determinar os pesos para cada fluxo. Este é o único número utilizado para determinar a quantidade de *bytes* que cada fila irá transmitir. Para determinar a alocação de banda para cada um dos fluxos basta dividir o peso do fluxo pelo total de *bytes* somando todos os fluxos. Por exemplo, se existem oito fluxos no sistema, cada um com um nível de precedência distinto, então cada fluxo terá peso igual ao nível de precedência +1 e a soma total será igual a:

1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36

Assim, o tráfego com nível de precedência 0 (zero) obterá 1/36 da banda e o tráfego com precedência 7 terá 8/36 da banda. Contudo, se em algum momento houver 7 fluxos com nível e precedência 2 e 1 fluxo de cada outro nível, então o total será:

1 + 2 + 3(7) + 4 + 5 + 6 + 7 + 8 = 54

Assim o tráfego com precedência 0 (zero) terá 1/54 da banda, cada fluxo de precedência 2 terá 3/54 da banda e assim por diante.

Como o roteador transmite um pacote inteiro (e não apenas alguns *bytes* dele), é necessário um mecanismo para fazer o escalonamento de pacotes, igualmente como é feito no CQ. Esse mecanismo conta o número de *bytes* virtualmente enviados dos pacotes para cada fila. Suponha a existência de 3 tipos de tráfego, o amarelo  com peso 1, o azul  com peso 2 e o laranja  com peso 3 e suponha ainda, que em um dado momento, cada uma das filas deseje transmitir um pacote. O contador de *bytes* irá transmitir (virtualmente) 1 *byte* do pacote amarelo, 2 *bytes* do azul e 3 do laranja em cada ciclo, depois a contagem recomeça novamente. Quando o último *byte* do pacote for virtualmente transmitido, então o pacote inteiro é enviado à interface de saída do roteador e assim transmitido.

Como ilustrado na figura 11, vê-se que o pacote laranja será transmitido primeiro, em seguida o amarelo e por último o azul. Observe que isso tudo é feito automaticamente sem a necessidade de uma prévia configuração conforme é feito no CQ [JC05].

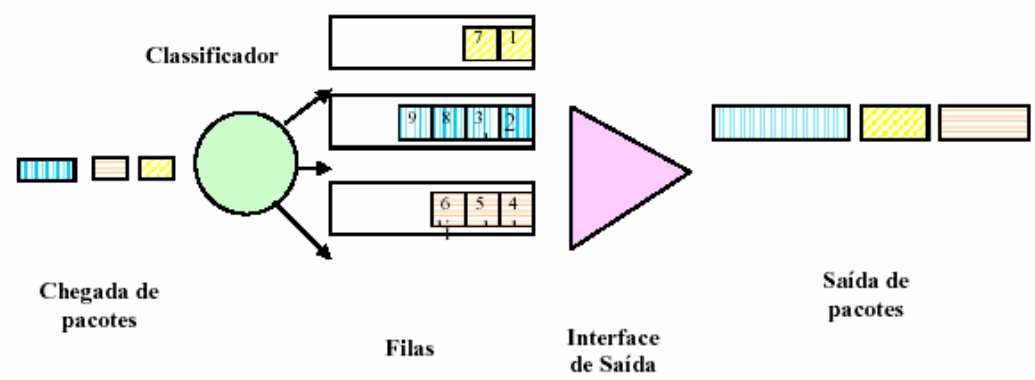


Figura 11 – Utilização de WFQ

3.1.4 **Protocolo de reserva RSVP**

Vamos considerar o caso de um transmissor e um receptor tentando fazer com que uma reserva de tráfego flua entre eles. Existem duas coisas que precisam acontecer antes que o receptor possa fazer a reserva. Primeiro, o receptor precisa saber que tráfego o transmissor provavelmente enviará, para que possa fazer uma reserva apropriada. Ou seja, ele precisa saber a TSpec do transmissor. Segundo, ele precisa saber que caminho os pacotes seguirão do transmissor ao receptor, para que possa estabelecer uma reserva de recursos em cada roteador no caminho. Esse dois requisitos podem ser atendidos pelo envio de uma mensagem do transmissor ao receptor, contendo a TSpec. A outra coisa que acontece é que cada roteador examina essa mensagem, chamada de mensagem PATH, enquanto passa e descobre o *caminho reverso* que será usado para enviar reservas do receptor de volta ao transmissor, em um esforço de levar a reserva a cada roteador no caminho.

Tendo recebido uma mensagem PATH, o receptor envia uma reserva de volta, em uma mensagem RESV. Essa mensagem contém a TSpec do transmissor e uma RSpec descrevendo os requisitos do receptor. Cada roteador no caminho examina a solicitação de reserva e tenta alocar os recursos necessários para satisfazê-lo. Se a reserva puder ser feita, a solicitação RESV é passada para o próximo roteador. Se não, uma mensagem de erro é retornada ao receptor que fez a solicitação. Se tudo correr bem, a reserva correta é instalada em cada roteador entre o transmissor e o receptor. Se o receptor quiser reter a reserva, ele enviará a mesma mensagem RESV aproximadamente uma vez a cada 30 segundos.

Poderemos verificar agora o que acontece quando um roteador ou enlace falha. Os protocolos de roteamento se adaptarão a falha e criarão um novo caminho

do transmissor ao receptor. As mensagens PATH são enviadas aproximadamente a cada 30 segundos, e podem ser enviadas mais cedo, se um roteador detectar uma mudança em sua tabela de encaminhamento, de modo que a primeira após a nova rota se estabilizar alcançará o receptor pelo novo caminho. A próxima mensagem RESV do receptor seguirá o novo caminho e, espera-se, estabelecerá uma nova reserva no novo caminho. Enquanto isso, os roteadores que não estão mais no caminho deixarão de receber as mensagens RESV, e essas reservas esgotarão seu tempo limite e serão liberadas[PD03].

3.2 O PADRÃO DIFFSERV

Em [ET01] descreve que na abordagem do modelo DiffServ uma porção do tráfego é tratada de forma privilegiada sobre o restante. É oferecida manipulação mais rápida, mais largura de banda na média e menor taxa de perda. Esta é uma preferência estatística, não uma garantia rígida. Com métodos adequados, incluindo a utilização de determinadas políticas nas extremidades da rede, a arquitetura de serviços diferenciados pode prover um tratamento adequado para uma boa gama de aplicações, incluindo aquelas de missão crítica, as que necessitam de baixo atraso, aplicações de telefonia IP. Na maioria das vezes, serviços diferenciados estão associados com a classificação do tráfego. O tráfego é agrupado em um pequeno número de classes e cada classe recebe uma Qualidade de Serviço na rede.

O grupo de trabalho DiffServ do IETF está trabalhando na especificação e definição de um padrão para os serviços de rede sob o nome genérico de Serviços Diferenciados. Este esforço está focado em grande parte no uso do campo TOS (Tipo de Serviço) do cabeçalho do IPv4, atualmente chamado campo DS (*Differentiated Service*), como um mecanismo de sinalização de QoS. A Figura 12 ilustra o uso deste campo. Os bits de 0 a 5 representam o DSCP (*Differentiated Service Code Point*), os bits 6 e 7 não são utilizados.

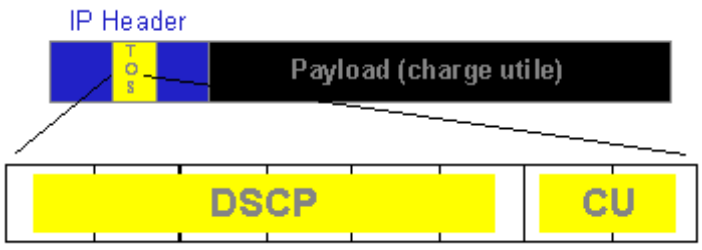


Figura 12 – Campo DSCP no Ipv4

Na arquitetura DiffServ não existe alocação explícita de recursos e não é feita sinalização, tendo em vista que a prioridade do pacote é transmitida no cabeçalho IP, permitindo desta forma maior escalabilidade e baixa sobrecarga de sinalização. Diffserv define o conceito de domínio DS, que é um conjunto contíguo de nós DS que aplicam um conjunto comum de políticas sobre o tráfego que atravessa o domínio. Um domínio DS tem nós de borda e nós de interior [RFC 2475] e [LL99]. Os nós DS de borda são responsáveis pela classificação e condicionamento do tráfego que entra no domínio DS [ET01].

Para cada fluxo de tráfego entrando no domínio pelos nós de borda, a política de QoS define qual terá um serviço diferenciado, como este deverá ser marcado nos nós de borda e como será tratado pelos nós interiores. Estes, por sua vez, examinam a marcação dos pacotes e atuam de acordo com as políticas definidas ou seu perfil de tráfego [ET01].

A arquitetura DiffServ define dois importantes componentes nos nós DS conforme mostra a Figura 13: os componentes de classificação e de condicionamento de tráfego.

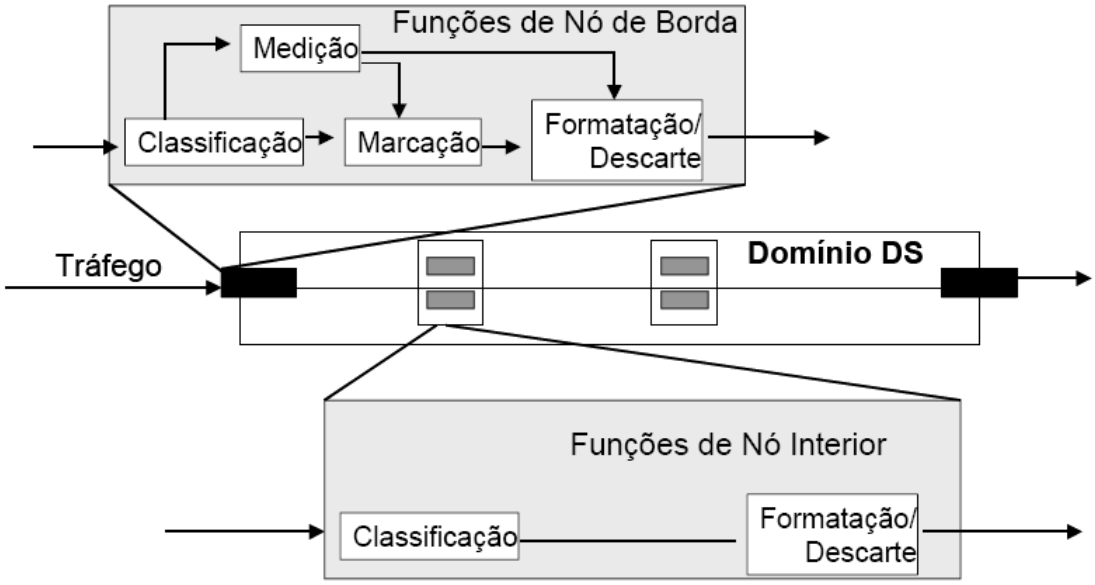


Figura 13 - Roteador que implementa arquitetura Diffserv

Existem dois tipos de classificadores: um que classifica o fluxo baseado apenas na classificação DS e outro que verifica múltiplos campos no cabeçalho IP. Estes classificadores são conhecidos como classificador de comportamento

agregado (BA *behaviour agregate*) e classificador multi-campo (MF . *multifield classifier*), respectivamente. A marca de classificação DS é conhecida como *DS codepoint* ou DSCP [ET01].

Em nós DiffServ, o *classificador* segundo [LL99] é o componente que divide o fluxo de entrada em um conjunto de fluxos de saída por meio de filtros de tráfego baseados no conteúdo do cabeçalho do pacote e/ou em diferentes atributos do pacote que podem ser implicitamente derivados.

De acordo com a chegada do pacote, o *medidor* verifica se o pacote está de acordo com um perfil de tráfego pré-definido. Dependendo da conformidade várias ações podem ser executadas, através dos *elementos de ações*. Três tipos de ações, e a combinação delas, podem ser executadas: *Marcação, formatação e descarte*.

Escalonamento é o processo de decidir no momento da transmissão qual fila entre o conjunto de candidatas deve ser servida, dependendo de alguma propriedade da fila e/ou do pacote, descrito anteriormente no item 3.1.3. A ação de *formatação* ou policiamento modifica o tráfego de entrada para forçar um determinado perfil de saída.

Condicionadores *de tráfego* são empregados em um determinado estágio do caminho dos dados para forçar uma determinada política. Eles podem ser implementados através da combinação de um ou mais componentes de DiffServ definidos anteriormente (classificadores, medidores, elementos de ação e filas) ou, alternativamente, através da combinação de condicionadores de tráfego existentes. Marcação é um exemplo de condicionamento de tráfego [ET01].

3.2.1 **Comportamento por salto – PHB**

Nos nós de borda, o fluxo de tráfego é classificado e marcado. Os campos DSCP (*DiffServ Code Point*) [RFC 1812] são mapeados para os PHBs (*Per Hop Behaviors*) [RFC 2474] definidos na arquitetura DiffServ. Os PHBs, [RFC 2597], [RFC 2598] definem o comportamento de encaminhamento de um pacote em um nó DiffServ.

PHBs são identificados através de um *label* de 6 bits, do campo TOS (*Type Of Service*) do cabeçalho do pacote IPv4 e o campo *Class* do cabeçalho do pacote IPv6 agora chamados de *Differentiated Services Code Point* (DSCP) conforme mostra a Figura 14.

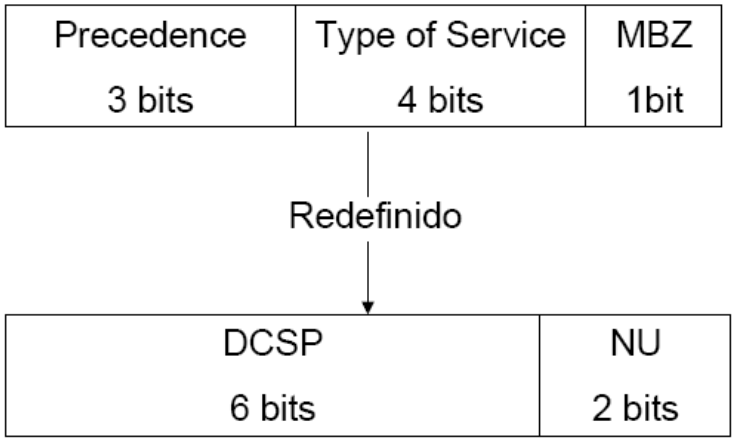


Figura 14 - Campo TOS redefinido para DSCP

Os seis bits do campo DS são usados para selecionar o PHB que o pacote terá em cada nó. Este campo é tratado como um índice em uma tabela usada para selecionar o mecanismo de manipulação de pacotes implementado em cada dispositivo. Este campo é definido como um campo não estruturado para facilitar a definição de futuros PHBs.

Outros comportamentos podem especificar que, para determinados pacotes, serão dadas certas prioridades relativas a outros, em termos de vazão média (*throughput)* ou de preferência para descarte, mas sem ênfase particular em atrasos. PHBs são implementados utilizando mecanismos de enfileiramento [ET01]. PHBs são comportamentos individuais aplicados em cada roteador, por isso isoladamente não garantem QoS fim-a-fim [MC99]. Entretanto, a interligação de roteadores com os mesmos PHBs e a limitação da taxa em que os pacotes são enviados para um PHB, possibilita o uso de PHBs para construir QoS de fim-a-fim. Por exemplo, a concatenação de EF PHBs ao longo de uma rota preestabelecida, com um cuidadoso controle de admissão, pode prover um serviço similar ao de uma linha dedicada, que é satisfatório para voz interativa [ET01].

O PHB *default* serve para o tráfego BE (*best effort*) melhor esforço. Ele assegura compatibilidade com o encaminhamento melhor esforço padrão em todos os roteadores [RFC1812] , [LL99]. PHBs de tráfegos com DSCP mais alto devem ter encaminhamento preferencial sobre aqueles com valor mais baixo.

Dois PHBs foram padronizados: o PHB EF (*Expedited Forwarding)* [RFC 2598] e o PHB AF (Assured Forwarding) [RFC 2597].

O PHB EF, também referido como serviço premium ou de canal dedicado, pode ser usado para tráfego com requisitos de baixa perda, baixo atraso, baixo jitter e garantia de largura de banda. Estes requisitos são alcançados assegurando-se que os agregados de tráfego encontram nenhum ou pouco enfileiramento.

As implementações devem prover meios de limitar o dano que o tráfego EF pode infligir sobre o outro tráfego. Os dispositivos de borda do domínio DS devem policiar o tráfego EF para assegurar a taxa de bits definida. Pacotes em excesso devem ser descartados.

Tabela 3.1 – Classes AF e precedência de descarte

Precedência de descarte	Classe AF1		Classe AF2		Classe AF3		Classe AF4	
	Binário	Decimal	Binário	Decimal	Binário	Decimal	Binário	Decimal
Baixa	001010	10	010010	18	011010	26	100010	34
Média	001100	12	010100	20	011100	28	100100	36
Alta	001110	14	010110	22	011110	30	100110	38

Tabela 3.2 – Classe Default – Tráfego melhor esforço

Precedência de descarte	Classe BE	
	Binário	Decimal
N/A	000000	0

Tabela 3.3 – Classe EF – Tráfego Premium

Precedência de descarte	Classe EF	
	Binário	Decimal
N/A	101110	46

O PHB AF tem por objetivo fornecer entrega de pacotes IP, com largura de banda assegurada, em quatro classes de transmissão, mas não oferece garantias quanto ao atraso. Cada classe tem três precedências de descartes, as quais são utilizadas para determinar a importância do pacote. Assim, um nó congestionado dá preferência para serem descartados, entre os pacotes de uma mesma classe, aqueles com maiores valores de precedência de descarte.

Os três primeiros bits do DSCP identificam a classe de transmissão, 001, 010, 011 e 100 e os três últimos bits definem a precedência de descarte, 010 para a precedência mais baixa, 100 para precedência média e 110 para a mais alta

precedência de descarte. Na Tabela 3.1, Tabela 3.2 e Tabela 3.3 são apresentados os DSCP recomendados para as classes de tráfego AF, BE e EF.

A marcação da precedência de descartes para tráfegos AF pode seguir dois enfoques: marcador com taxa simples [RFC 2697] e marcador com taxa dupla [RFC 2698]. Ambos usam um regulador do tipo duplo balde furado (*Dual Leaky Bucket*), sendo que no marcador simples, os dois baldes são preenchidos na mesma taxa e no marcador com taxa dupla, os baldes são preenchidos com duas taxas diferentes. Ambos os enfoques medem o tráfego e marcam o pacote, dependendo se o fluxo excede a taxa acordada ou contratada CIR (*Committed Information Rate*). Um pacote é marcado como *verde*, *amarelo* ou *vermelho* dependendo em quanto o CIR foi excedido. Um pacote *verde* tem a menor prioridade de descarte, enquanto que um pacote *vermelho*, a mais alta. O principal objetivo é fornecer a marcação para algum mecanismo de descarte baseado na precedência de descartes.

Em [ET01] descreve que arquitetura DiffServ *serviço* está associado à QoS do ponto de vista da aplicação, permitindo especificar as necessidades destas em termos de largura de banda, atraso, *jitter* e taxa de perdas, e pode ser quantitativo ou qualitativo. No primeiro caso, o *serviço* é especificado através de um conjunto de métricas e valores de referência correspondentes, enquanto que no segundo caso somente uma definição de alto nível é fornecida. Em ambos os casos a implementação de QoS está baseada em uma determinada combinação dos componentes de DiffServ acima mencionados. Isto faz a arquitetura de DiffServ muito flexível.

Quanto aos serviços oferecidos por um domínio aderente a DS, devemos notar:

- serviços DS são todos para tráfego unidirecional apenas; e
- serviços DS são para tráfegos agregados, não fluxos individuais. Em [RFC 2475] um serviço é definido como o tratamento global de um subconjunto do tráfego do cliente dentro de um domínio aderente a DS ou fim-a-fim. O tráfego na rede geralmente atravessa uma concatenação de redes que podem incluir *hosts*, redes residenciais e de escritório, redes corporativas/campus e várias redes de longa distância. Redes residenciais e de escritório são normalmente clientes de redes de campus ou corporativas, que são por sua vez clientes de redes de longa distância.

Os clientes podem marcar os campos DS dos pacotes para indicar o serviço desejado, ou estes campos podem ser marcados pelo roteador de borda que liga o cliente à rede, baseado na classificação MF. No ponto de ingresso da rede, os pacotes são classificados, policiados e, possivelmente, atrasados para torná-los aderentes a algum perfil de tráfego pré-definido. As regras de classificação, policiamento e atrasos usadas nos roteadores de ingresso da rede são derivadas a partir de um Acordo de Nível de Serviço (SLA . *Service Level Agreement*) [ST99]. O montante de espaço de bufferização necessário para estas operações também é derivado dos SLAs.

3.3 INTERLIGAÇÃO DOS MODELOS DIFFSER E INTSERV

A interligação dos modelos IntServ e DiffServ tem por principal objetivo o suporte de QoS às aplicações, fim-a-fim, num cenário em que as redes periféricas utilizem IntServ e as redes de *core* utilizem DiffServ. Neste cenário a escalabilidade do modelo DiffServ ajuda a estender e a generalizar a maior funcionalidade do modelo IntServ [RFC3175]. Os domínios DiffServ intermediários neste cenário de interligação são vistos pelos domínios IntServ como ligações virtuais, ou túneis, entre os últimos.

No cenário descrito, os sistemas terminais e roteadores que constituem as redes periféricas IntServ sinalizam entre si, de uma forma transparente, os pedidos de reserva de recursos através da rede DiffServ. Nas fronteiras entre os dois domínios as mensagens RSVP são processadas e submetidas a um controle de admissão baseado na disponibilidade dos recursos apropriados dentro da rede DiffServ.

A figura 15 ilustra o cenário de interligação IntServ-DiffServ. Neste cenário, o estabelecimento do fluxo de comunicação fim-a-fim ocorre da seguinte maneira:

- O emissor, localizado numa rede IntServ, gera uma mensagem RSVP PATH, destinada ao receptor localizado na rede IntServ remota, identificando e caracterizando o tráfego da sua aplicação;
- Ao atravessar as redes periféricas IntServ a mensagem RSVP PATH é processada normalmente, na região DiffServ a mensagem é transportada de uma forma transparente;
- Quando a mensagem chega ao receptor é processada dando origem a uma mensagem RSVP RESV com a especificação da QoS pretendida;

- A mensagem RSVP RESV é enviada com destino ao emissor através da rede DiffServ, podendo vir a ser rejeitada nas regiões IntServ caso os recursos sejam insuficientes para satisfazer a QoS pretendida;
- Quando a mensagem RSVP RESV chega ao *edge router 1* da região IntServ onde está localizado o emissor, os mecanismos de controle de admissão deverão atuar com base nos recursos especificados no *Traffic Conditioning Agreement* (TCA);
- Após o controle de admissão deverão atuar os mecanismos de mapeamento do serviço IntServ pretendido, com as características de QoS especificadas, na classe de serviço DiffServ;
- Após a atuação dos mecanismos de mapeamento a mensagem RSVP RESV é encaminhada para o emissor na rede IntServ, sendo atualizada, no *edge router 1,* a entrada correspondente à classe de serviço Diffserv mapeada no fluxo IntServ.
- Durante a comunicação, ou enquanto as reservas RSVP forem renovadas com sucesso, o *edge router 1* marca os pacotes do fluxo IntServ com o DSCP correspondente à CoS DiffServ atribuída, recebendo assim o serviço apropriado da rede DiffServ.

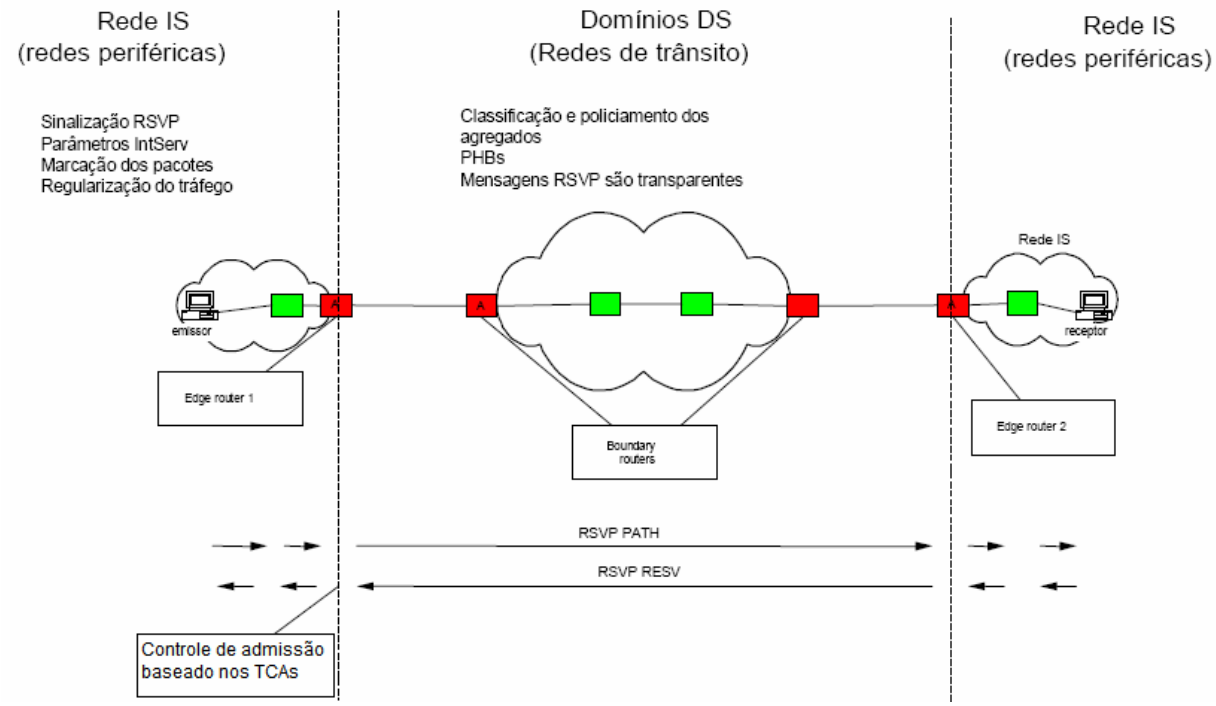


Figura 15 – Interligação de redes IntServ através de redes DiffServ

No cenário descrito, o funcionamento dos mecanismos de mapeamento e controle de admissão é crucial para a manutenção da QoS fim-a-fim aos fluxos das aplicações.

3.4 POLICIAMENTO E CONTROLE DE TRÁFEGO

Técnicas de controle de congestionamento monitoram o tráfego na rede no sentido de antecipar e evitar a ocorrência de congestionamento, usualmente através do descarte de pacotes. As duas técnicas mais utilizadas para este fim são a RED e WRED (*Random Early Detection* e *Weighted Random Early Detection* respectivamente).

3.4.1 **RED - Random Early Dectection**

Quando ocorre um *timeout* no TCP transmissor, o protocolo reduz o tamanho da janela de transmissão e inicia o processo de partida lenta, onde o tamanho cresce gradativamente à medida que o transmissor vai recebendo reconhecimentos positivos do receptor.

Se um pacote de um fluxo é descartado pela rede, ocorrerá um timeout e o procedimento anteriormente descrito têm início. Como conseqüência da redução do tamanho da janela de transmissão, temos a redução na taxa de transmissão de pacotes.

Se a perda de pacote é devida a congestionamento no roteador, a redução da taxa de transmissão de pacotes por parte do transmissor, resultante desta perda, irá aliviar a situação de congestionamento. Se a situação de congestionamento leva ao descarte de pacotes de vários fluxos distintos, teremos vários transmissores reduzindo suas janelas de transmissão e iniciando o processo de partida lenta, eliminando o congestionamento. Porém, estes transmissores aumentarão suas janelas de transmissão conjuntamente, e conseqüentemente, teremos novamente uma situação de congestionamento, que resultará em novos descartes de pacotes e no reinicio do processo. Este ciclo é conhecido como Problema de Sincronização Global. O algoritmo RED tenta evitar esse problema atuando de forma preventiva ao congestionamento, ou seja, o RED tenta evitar que o congestionamento ocorra.

No RED, quando uma situação de tendência de congestionamento é detectada (o tamanho da fila ultrapassar um determinado limiar), inicia-se um

processo de descarte aleatório de pacotes, onde a probabilidade de descarte é função da taxa de ocupação da fila. Este descarte antecipado irá resultar na diminuição da taxa de chegada de pacotes no roteador, devido ao mecanismo de operação do TCP.

O RED só funciona adequadamente em conjunto com protocolos de transporte que sejam robustos quanto à perda de pacotes, como TCP. Se o protocolo de transporte não reage à perda de pacotes com a diminuição da taxa de transmissão, o RED não terá nenhum efeito positivo, podendo inclusive deteriorar o desempenho do sistema pelo aumento da taxa de perda dos pacotes.

3.4.2 **WRED - Weighted Random Early Dectection**

No algoritmo WRED a probabilidade de um pacote entrante ser descartado é definida pela taxa de ocupação da fila e por um peso associado ao fluxo ao qual o pacote pertence. O que se busca com o WRED é que pacotes com maior prioridade tenham menor probabilidade de descarte . Por exemplo, uma probabilidade de descarte menor pode ser associada a fluxos de pacotes com maior prioridade (determinada pelo conteúdo dos bits de precedência do campo TOS do IP), ou fluxos de pacotes com reserva de recursos (através do RSVP) a figura 16 mostra um exemplo de funcionamento.

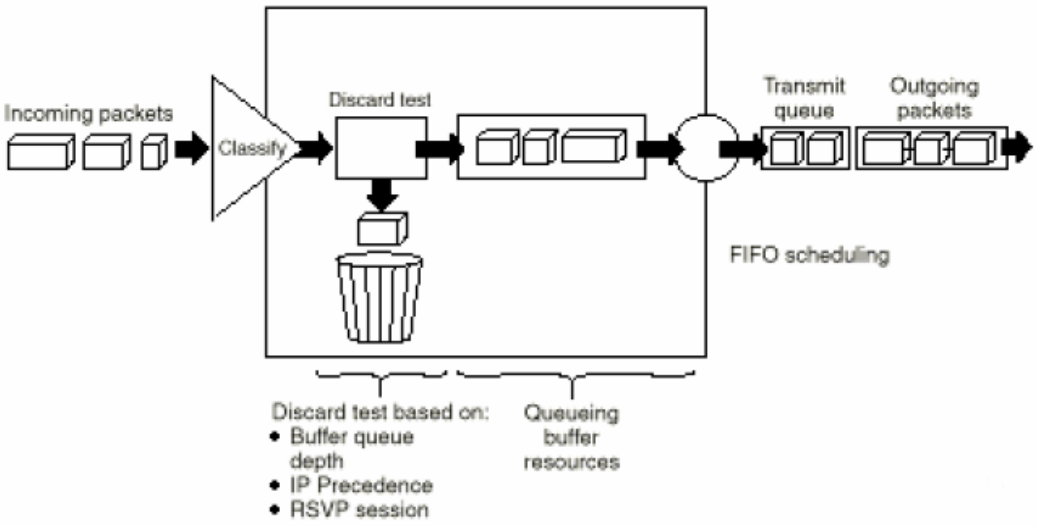


Figura 16 - Operação do algoritmo WRED

O uso do algoritmo WRED não resulta na priorização estrita que o tráfego de voz requer. Porém, o WRED pode prover um tratamento preferencial aos pacotes de

voz durante situações de congestionamento, minimizando a perda destes pacotes pelo descarte antecipado de pacotes de dados, aos quais se atribui uma maior probabilidade de descarte [CIS].

Deve-se lembrar que o descarte de um pacote de voz não reduzirá o fluxo de chegada deste tipo de pacote, uma vez que o UDP não reage à perda de pacotes. Portanto, um fluxo muito pesado de tráfego de voz pode causar um overflow em uma fila WRED e conseqüentemente uma elevada taxa de perda de pacotes. Ainda, se a probabilidade de descarte, pelo WRED, associada a pacotes de voz não for muito baixa, podemos ter uma taxa de perda de pacotes inaceitável para este tipo de tráfego e a conseqüente perda do nível de QoS.

3.4.3 **Policiamento e Conformação de Tráfego**

As funções de policiamento e conformação usualmente identificam as violações no tráfego da rede de uma mesma maneira. Elas diferem, contudo, na forma como elas respondem a essas vialações, por exemplo [QSS]:

- A função de policiamento usualmente descarta o tráfego que não está conforme ou o define como elegível para descarte.
- A função de conformação tipicamente atrasa o tráfego em excesso, através de mecanismos de enfileiramento, retendo os pacotes e liberando-os de maneira tal que o fluxo de saída esteja dentro dos parâmetros definido.

A técnica Token Bucket, é a mais comumente utilizada para as funções de policiamento e conformação de tráfego. Alguns algoritmos implementados pela Cisco se baseiam no Token Bucket, como por exemplo, o CAR, GTS e FRTS

O Token Bucket é uma definição formal de uma taxa de transferência. Ele possui três componentes:

- Comprimento de rajada: especifica a máxima rajada em que pode ser enviada dentro de um intervalo de tempo;
- Taxa média: especifica quantos bits podem ser enviados por unidade de tempo, em média. Por definição, sobre qualquer múltiplo inteiro do intervalo, a taxa de bits da interface não excederá a taxa média. Dentro de um intervalo a taxa de bit pode exceder momentaneamente a taxa média;
- Intervalo de tempo: também chamado de intervalo de medida, especifica

o intervalo de tempo em que se define o comprimento de rajada e a taxa média.

O algoritmo Token Bucket, mostrado na figura 17, opera da seguinte maneira:

- Fichas são depositadas em um balde, com capacidade para C fichas, a uma taxa constante (CIR);
- Se o balde enche de fichas, as próximas fichas são descartadas;
- A transmissão de pacote consome do balde uma quantidade de fichas igual ao tamanho do pacote em bytes;
- Quando chega um pacote e não há fichas em quantidade suficiente no balde, o pacote é declarado não conforme e uma das duas opções podem ser tomadas:
    i. O pacote é descartado ou definido como elegível para descarte;
    ii. O pacote é atrasado até que se tenha fichas suficientes no balde;
- Quando não se tem pacotes para transmitir, as fichas acumulam-se no balde até atingir sua capacidade C, permitindo que se transmita posteriormente rajadas de pacotes.

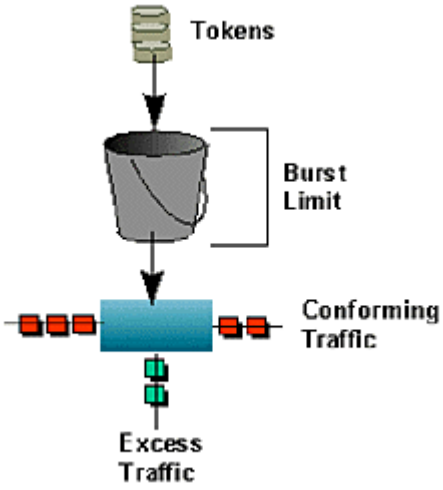


Figura 17 – Funcionamento do Token Bucket

A maior rajada de dados que se pode transmitir ocorre quando o balde está cheio e chega um pacote. O tamanho desta rajada é definido pela capacidade do balde mais o intervalo de tempo dividido pela taxa de chegada de fichas no balde. Em regime permanente, a taxa de transmissão de dados não excede a taxa de chegada de fichas no balde.

4 **CONCLUSÃO**

Este trabalho teve como objetivo o estudo do comportamento do tráfego de voz sobre uma rede IP, onde foram analisadas várias técnicas de transmissão usando os padrões IntServ e Diffserv. Hoje a transmissão de sinais de voz pela Internet é uma realidade, concluindo que o assunto é bastante pertinente no contexto atual, pois a perda de pacotes em redes IP é bastante significativa, o que pode tornar problemática a manutenção de um nível aceitável de qualidade na transmissão de voz pela Internet. No apêndice A deste trabalho são mostradas várias técnicas de QoS aplicadas em tráfegos de voz sobre IP, através de um ambiente de simulação montado em roteador cisco e sistemas Linux. Neste estudo, os gráficos mostrados indicam o impacto dos parâmetros de QoS sobre a qualidade das chamadas VoIP, comprovando que uma melhora significativa na qualidade de voz pode ser obtida através do emprego dessas técnicas.

REFERÊNCIAS BIBLIOGRÁFICAS

[JC05] MAGRO, Júlio César. Estudo da Qualidade de Voz em Redes IP. Dissertação de Mestrado, julho 2005, Faculdade de Engenharia Elétrica e de Computação FEEC/UNICAMP.

[ET01] LOPES, Edson Tadeu. Qualidade de Serviço em Redes IP com DiffSer: Avaliação através de medições. Dissertação de Mestrado, maio de 2001, Universidade Federal de Santa Catarina.

[SC01] LUNARDI, Sediane Carmem. Uma Camada de Suporte à Quailidade de Serviço para Aplicações Multimídia. Dissertação de Mestrado, 2001

[DGS03] SILVA, Davison Gonzaga da. Implementação de Um Sistema SIP Para o Sistema Operacional Linux. Dissertação de Mestrado, dezembro 2003, UNICAMP.

[SCH98b] SCHULZRINNE, Henning, RAO, A., LANPHIER, R. “Real Time Streaming Protocol (RTSP)”, Internet RFC 2326, Internet Engineering Task Force, April 1998.

[TI02] HERSENT, Oliver. GURLE, David. PETIT, Jean-Pierre. Telefonia IP. Comunição Multimídia Baseada em Pacote. Addison Wesley, 2002.

[AT96] ATM User-Network Interface(UNI; Signaling Specification – version 4.0); Publicação do ATM Fórum, 1996.

[FJ99] FIGUEIREDO JR, Themistocles. Um serviço de Garantia Estatística do Atraso em Redes ATM para Aplicações Multimídia.

[DI98] DINIZ, A. Um Serviço de Alocação Dinâmica de Banda Passante em Redes ATM para Suporte a Aplicações Multimídia; Dissertação de Mestrado, março de 1998, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais.

[HA95] HAFID, A. *Et al*. On Quality of Service Negotiation for Distributed Multimedia Applications; Publication Departamental 977, Université de Montreal, 1995.

[LI98] LIMA, F. Um Modelo de Qualidade de Serviço baseado em ATM para a Plataforma Multware; Dissertação (Mestrado), 1998, Universidade Estadual de Campinas.

[WH97] WHITE, Paul. RSVP and Integrated Services in the Internet: A Tutorial, IEEE Communications Magazine, maio 1997, p. 100-106.

[ML01] MELO, Edson Tadeu Lopes. Qualidade de Serviço em Redes IP com DiffServ: Avaliação Através de Medições; Dissertação (Mestrado), maio 2001, Universidade Federal de Santa Catarina.

[DG04] COSTA, Daniel Gouveia. Uma Introdução ao Padrão ITU H.323. Rede Nacional de Ensino e Pesquisa, novembro 2004, RNP.

[RFC 3550] Schulzrinne, H.; Casner, S.; Frederick, R.; Jacobson, V.; (2003): *A Transport Protocol for Real-Time Applications.* Request for Comments 3550.

[RFC 1633] Braden, R.; Clark, D. & Shenker, S. (1994): *Integrated Services in the Internet Architecture: an Overview,* Request for Comments 1633.

[RFC 2475] Black, D.; Blake, S.; Carlson, M.; Davies, E.; Wang, Z. & Weiss, W. (1998): *An Architecture for Differentiated Services*, Request for Comments 2475.

[RFC 1812] Baker, F. (Ed) (1995): *Requirements for IP Version 4 Routers*, Request for Comments 1812.

[RFC 2597] Heinanen, J.; Baker, F.; Weiss, W. & Wroclawski, J. (1999): *Assured Forwarding PHB Group*, Request for Comments 2597.

[RFC 2598] Jacobson, V.; Nichols, K.&. Poduri, K. (1999): *An Expedited Forwarding PHB*, Request for Comments 2598.

[RFC 2597] Heinanen, J.; Baker, F.; Weiss, W. & Wroclawski, J. (1999): *Assured Forwarding PHB Group*, Request for Comments 2597.

[RFC 2697] Heinanen, J. & Guerin, R. (1999): *A Single Rate Three Color Marker,* Request for Comments 2697.

[RFC 2698] Heinanen, J. & Guerin, R. (1999): *A Two Rate Three Color Marker* , Request for Comments 2698.

[RFC 2474] Nichols, K.; Blake, S.; Baker, F. & Black, D. (1998): *Definition of the Differentiated Services Filed (DS Field) in the Ipv4 and Ipv6 Headers,* Request for Comments 2474.

[LL99] LINNEY, L. (1999): Differentiated Services on IBM 221x Routers, IBM Co.

[PD03] PETERSON, Larry L. e DAVIE, Bruce S.. Redes de Computadores. Uma abordagem de sistemas. Tradução da terceira edição. Editora Campus, 2003.

[Sch2000] Schmidt, Ana Luisa Pereira. O Protocolo RSVP e o desempenho de Aplicações Multimídia. Boletim bimestral sobre tecnologias de rede publicado pela RNP, 12 de maio de 2000. http://www.rnp.br/newsgen/0005/rsvp.html. Acessada em outubro/2006.

[MC99] Microsoft Corporation (1999): *Quality of Service Technical White Paper*. http://msdn.microsoft.com/library/psdk/gqos/qosstart_2cdh.htm. Acessada em outubro/2006.

[ST99] Stardust Technologies, Inc. (1999): *Introduction to QoS Policies*, White Paper. http://www.qosforum.com/white-papers/qospol_v11.pdf. Acessada em novembro/2006

[THO96] THOMAS, Stephen A. IPng and TCP/IP protocols: implementing the next generation internet. New York: John Wiley, 1996. 481 p.

[BUS96] BUSSE, Ingo, DEFFNER, B., SCHULZRINNE, H. “Dynamic QoS Control of Multimedia Applications based on RTP”, Computer Communications, [s.l.], vol 19, p. 49-58, Jan. 1996.

[RFC 1890] SCHULZRINNE, H. RTP *Profile for Audio and Video Conferences with Minimal Control*. RFC 1890, IETF Audio-Video Transport Working Group. Jan. 1996.

[RFC 1889] SCHULZRINNE, H. et al. RTP: *A Transport Protocol for Real-Time Applicatons.* RFC 1889, IETF Audio-Video Transport Working Group. Jan. 1996.

[SME04] SMETANA, George M. M. Arcuri. Um Sistema de Conferência Centralizada com Controle de Posse da Palavra para Educação a Distância. Dissertação de Mestrado. Escola Politécnica da USP. 2004.

[CIS] Quality of Service for Voice Over IP Solutions Guide. CISCO SYSTEMS. Documento disponível em www.cisco.com. Acessada em novembro/2006

[QSS] Quality of Service Solutions Cofiguration Guide, Police and Shaping Overview. Documento elaborado pela Cisco. Disponível em www.cisco.com. Acessada em novembro/2006

[RFC 3175] F. Baker et al, Aggregation of RSVP for IPv4 and IPv6 Reservations, RFC3175, IETF, Setembro, 2001.

ANEXO I

Os gráficos abaixo mostram as várias técnicas de enfileiramento usadas para estabelecer uma qualidade de serviço ao tráfego de voz. Essas técnicas foram utilizadas em roteadores cisco e sistemas Linux.

1- Neste primeiro gráfico é empregado a política de fila FIFO com link de 500 kbps sem tráfego de fundo.

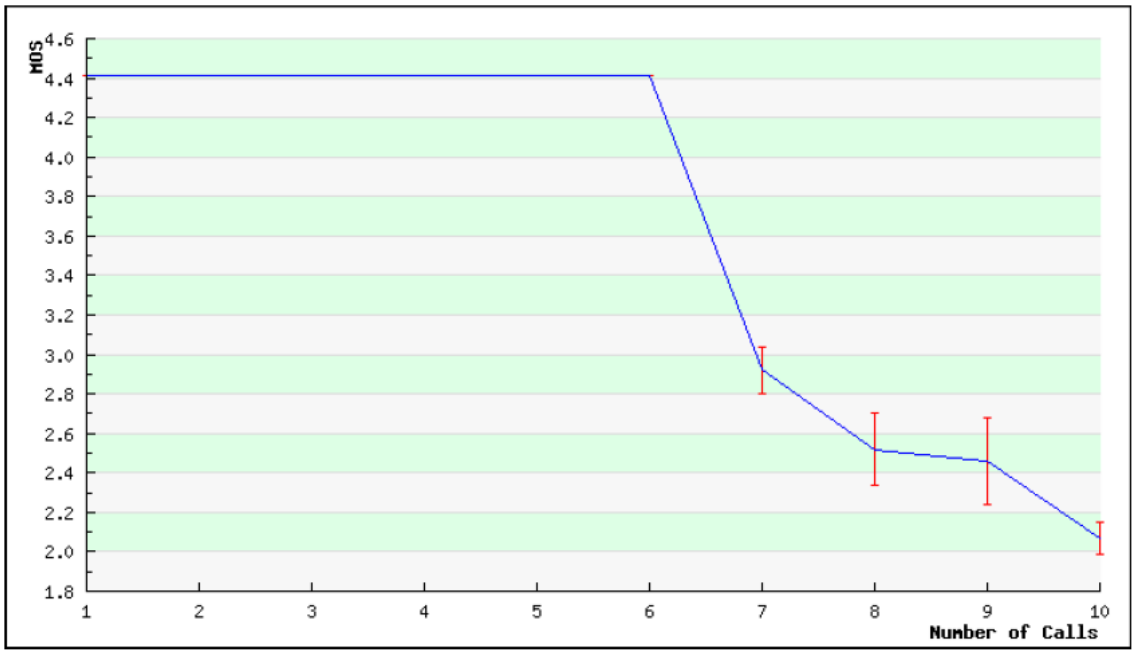


Figura 18 – Política FIFO sem tráfego de fundo

O emprego do enfileiramento FIFO foi em um roteador cisco, como visto em um link de 500 kbps e sem tráfego de fundo foi possível realizar 6 chamadas com um MOS de 4,4 , isto é, com uma boa qualidade.

Nesse mesmo exemplo, o gráfico abaixo mostra como o tráfego de fundo de 200 kbps pode alterar o número de chamadas com boa qualidade, utilizando o enfileiramento FIFO.

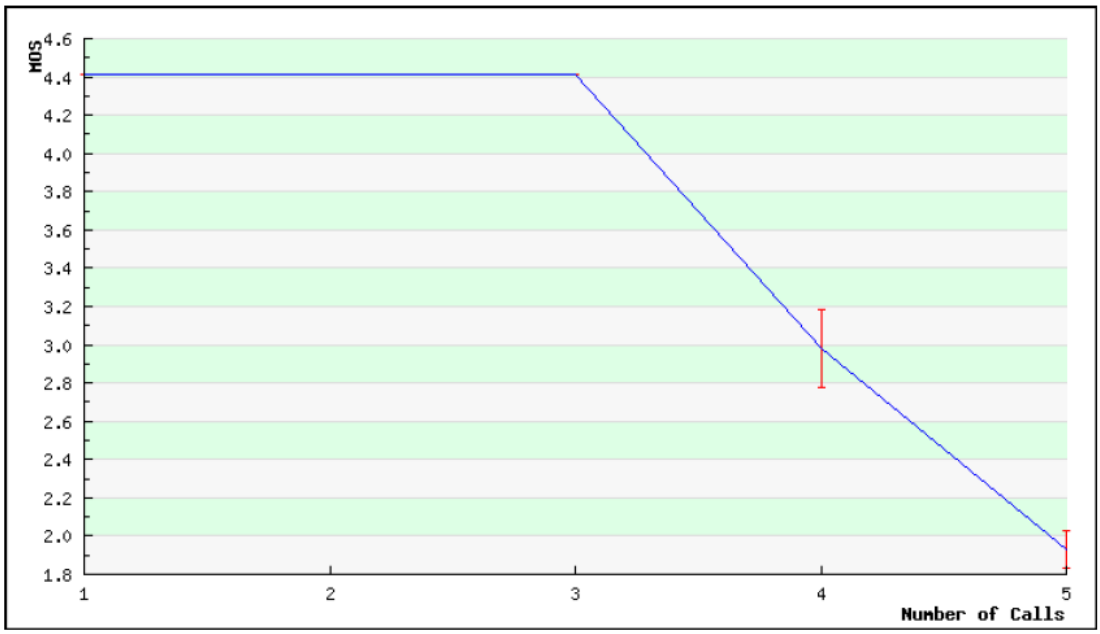


Figura 19 – Política FIFO com tráfego de fundo de 200 kbps

Agora utilizando um tráfego de fundo de 400 kbps o gráfico abaixo mostra que apenas uma chamada pode ser feita.

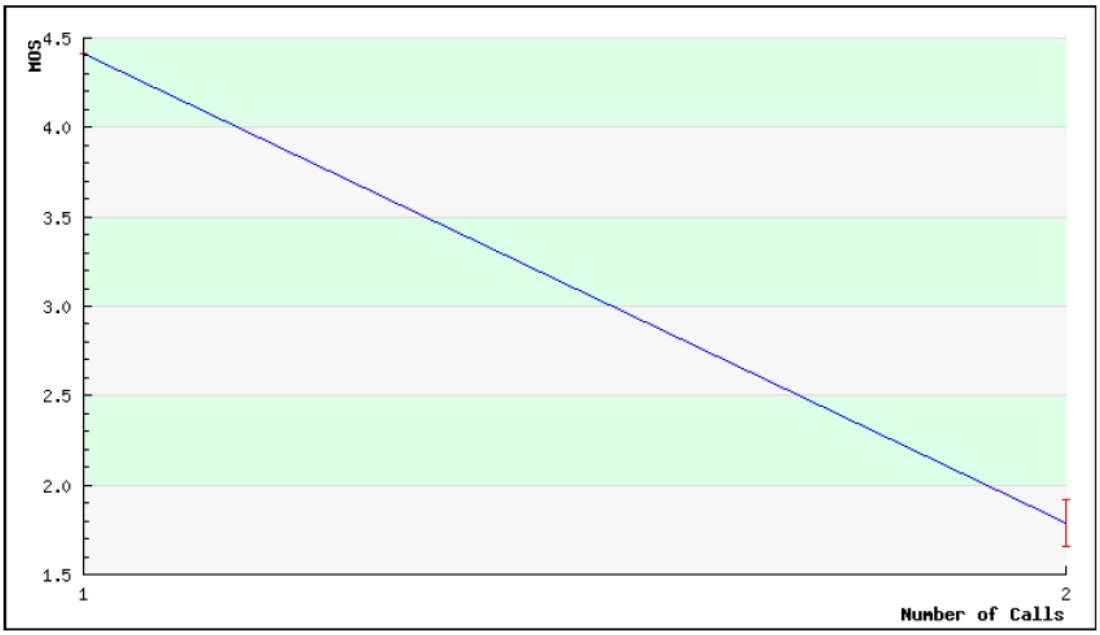


Figura 20 – Política FIFO com tráfego de fundo de 400 kbps

2 – Agora aplicamos a política WFQ para o enfileiramento.

Utilizando um link ainda de 500 kbps e tráfego de fundo de 200 kbps notamos no gráfico que foram realizadas 5 chamadas com boa qualidade.

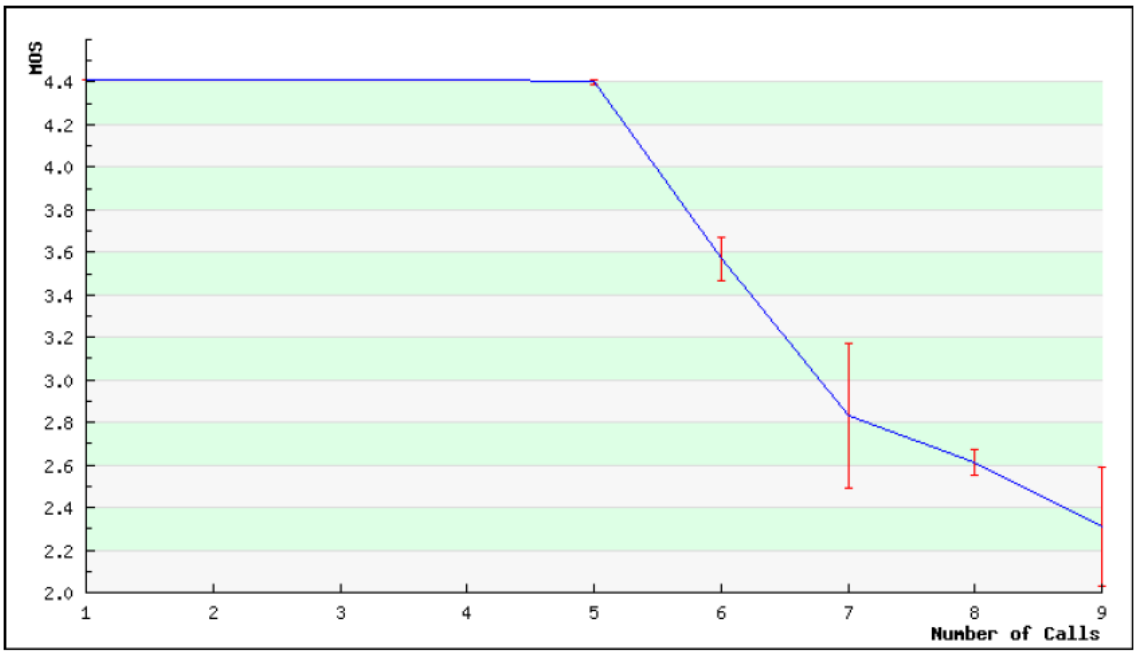


Figura 21 – Política WFQ tráfego de fundo 200 kbps

Neste outro notamos, ao aplicar um tráfego de fundo de 400 kbps, ainda foi possível realizar 5 chamadas com boa qualidade.

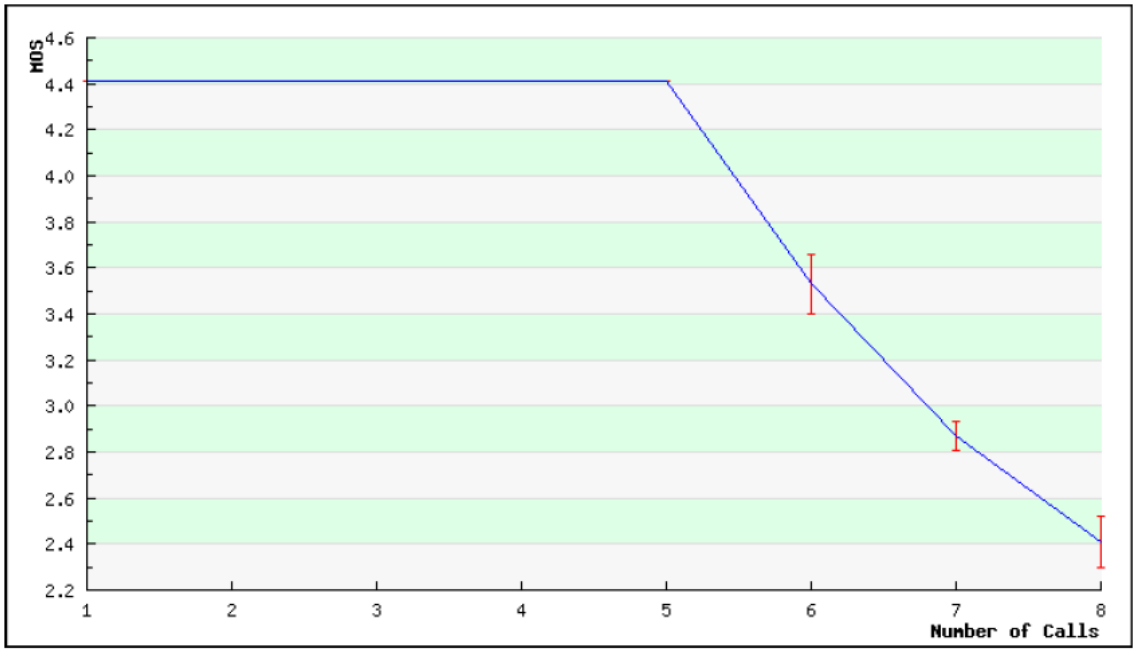


Figura 22 – Política WFQ com tráfego de fundo de 400 kbps

Em um último exemplo usando WFQ com tráfego de fundo de 800 kbps verificamos que foi possível realizar 5 chamadas com uma boa qualidade.

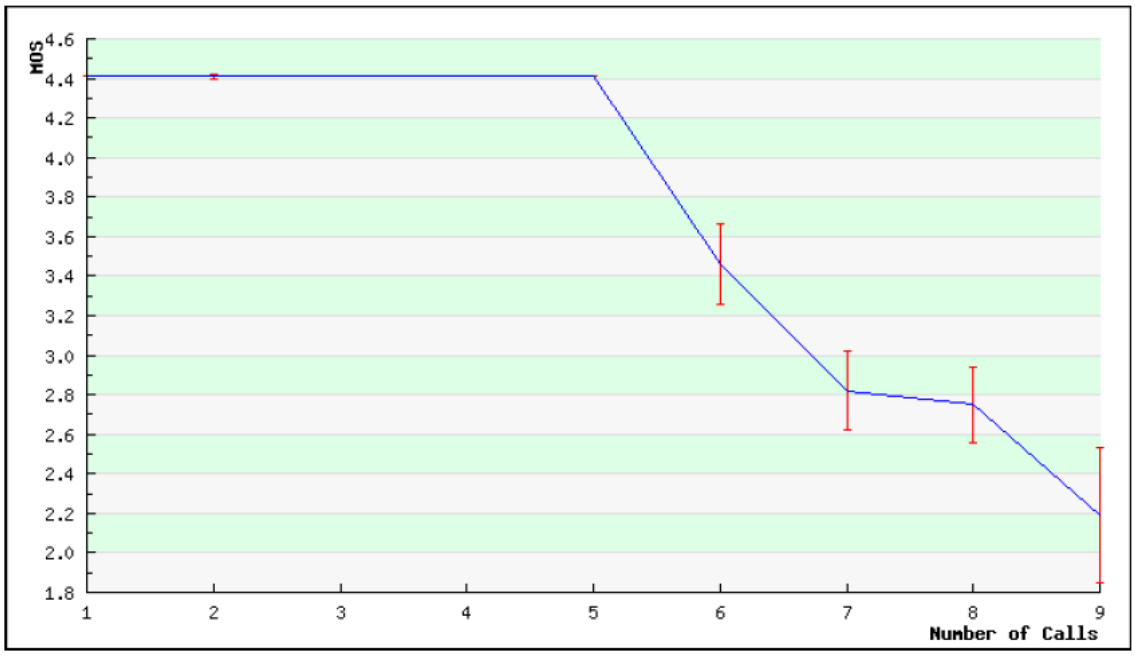


Figura 23 – Política WFQ com tráfego de fundo de 800 kbps

Isso mostra que ao utilizar uma política WFQ, podemos resolver as limitações encontradas na política FIFO, impedindo que uma rajada de um tipo de tráfego monopolize toda a banda disponível. Utilizando pesos para identificar e classificar o tráfego de voz e determinar quanto de banda esse tráfego pode possuir em relação a outros.

3 – Neste exemplo foi usado a política PQ de 320 kbps para tráfego de voz marcado com DSCP EF e o tráfego de sinalização de voz marcado com DSCP AF31 em um roteador cisco.

Sem tráfego de fundo

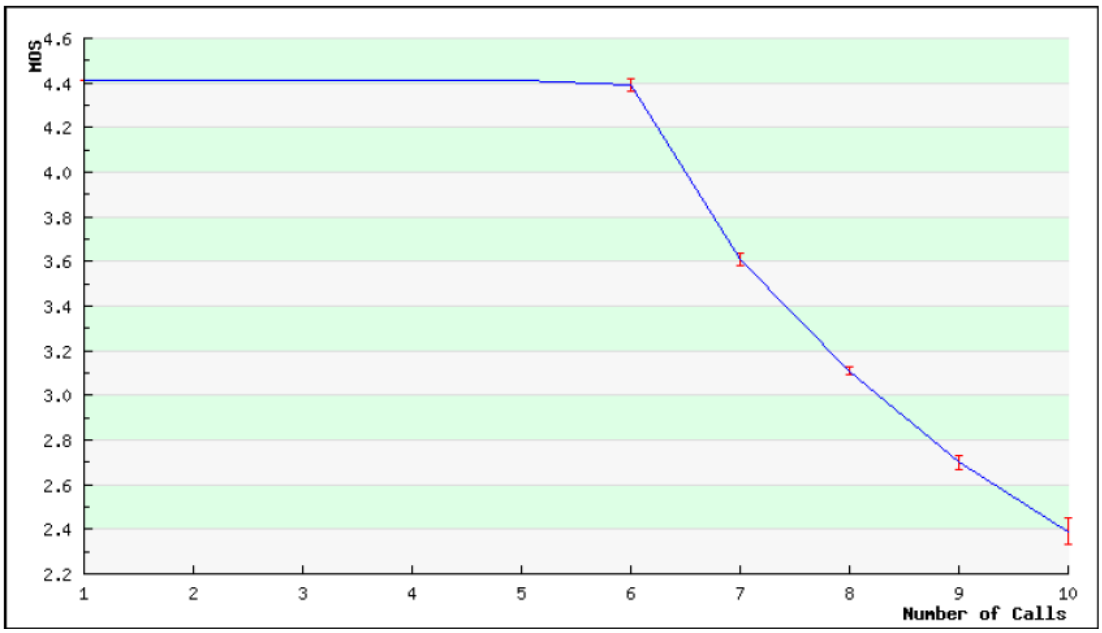


Figura 24 – Política PQ sem tráfego de fundo

Com tráfego de fundo de 400 kbps

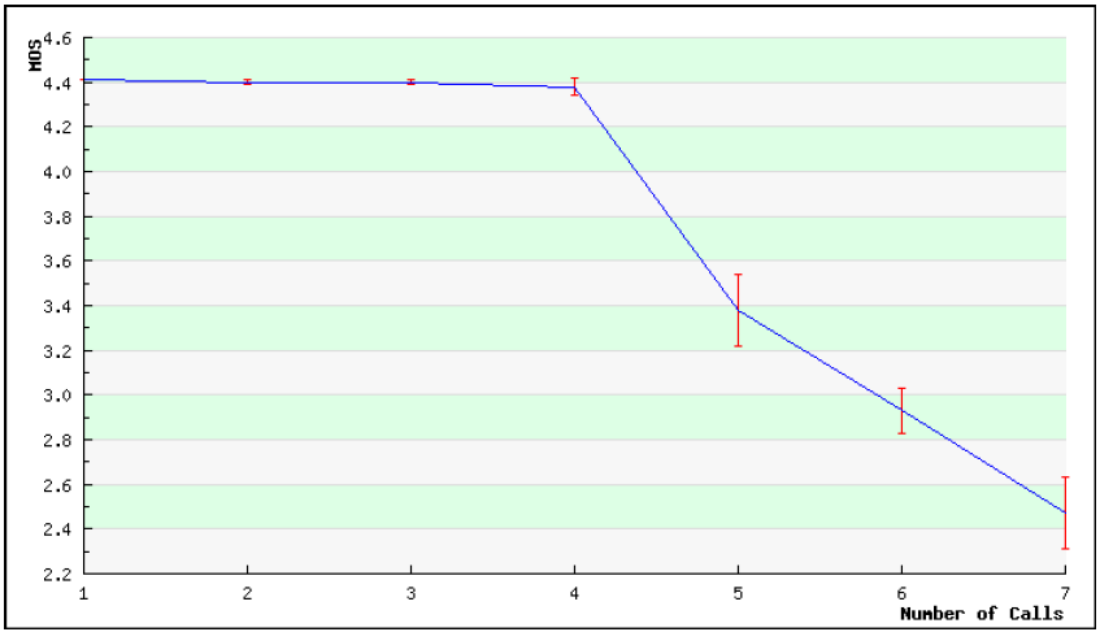


Figura 25 – Política PQ com tráfego de fundo de 400 kbps

Com tráfego de fundo de 200 kbps

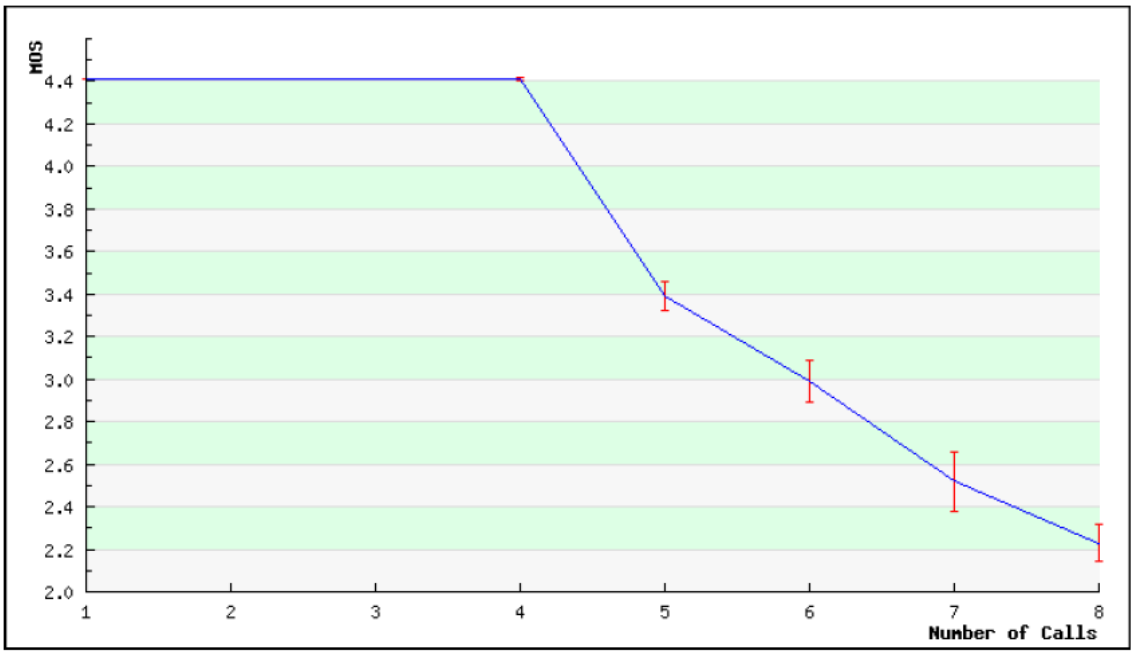


Figura 26 – Política PQ com tráfego de fundo de 200 kbps

Com tráfego de fundo de 800 kbps

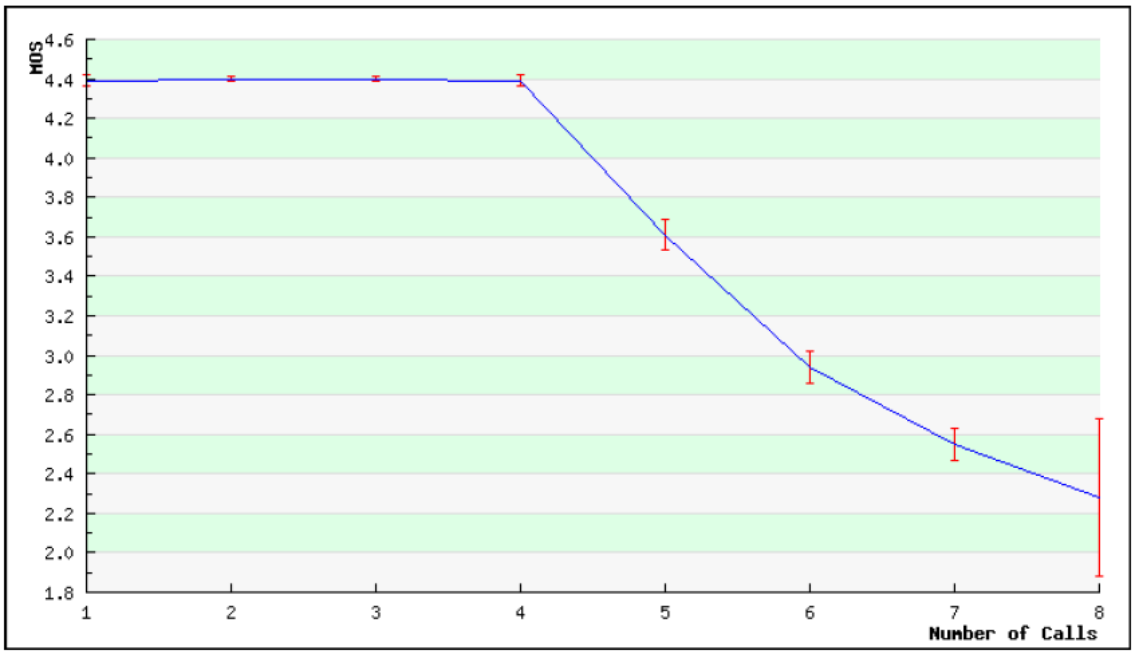


Figura 27 – Política PQ com tráfego de fundo de 800 kbps

Com uma política de PQ aplicada no tráfego e sinalização de voz, vimos que mesmo com um tráfego de fundo alto os pacotes de voz são priorizados e enviados até que não haja mais pacotes de voz a serem enviados, somente depois os pacotes de dados são enviados.

4 – No exemplo abaixo foi utilizado um roteador Linux com enlace de 100 Mbps com filtro de 500 Kbps.

Política FIFO com 500 Kbps de tráfego de fundo.

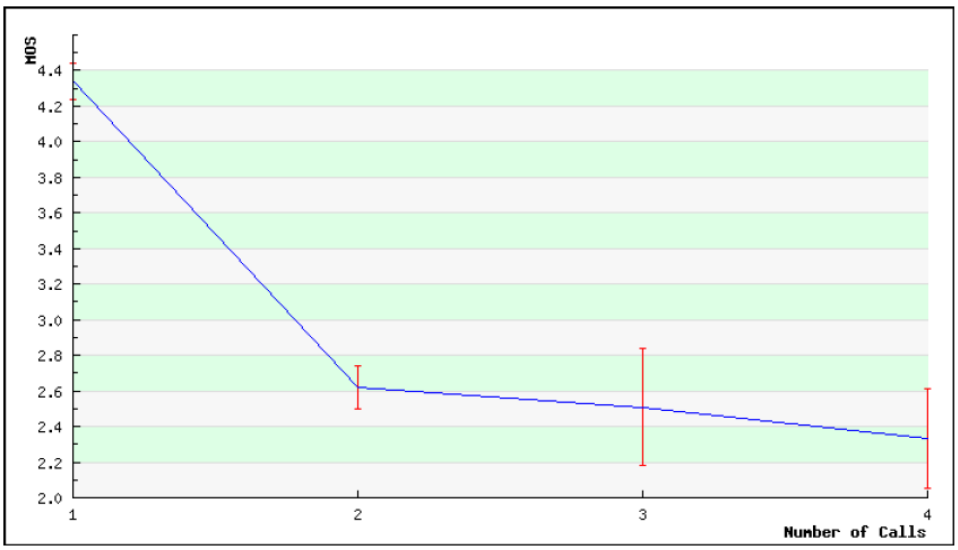


Figura 28 – Política FIFO Linux com tráfego de fundo de 500 kbps

Política SFQ (100 pacotes de buffer) tráfego de fundo de 500 kbps.

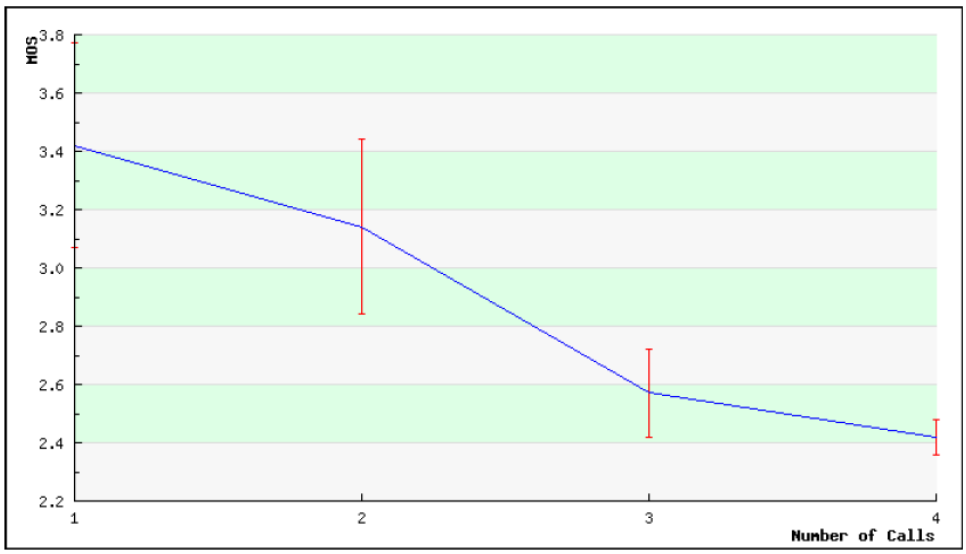


Figura 29 – Política SFQ em Linux com buffer de 100 PKT's

Política SFQ (16 pacotes de buffer) tráfego de fundo de 500 kbps

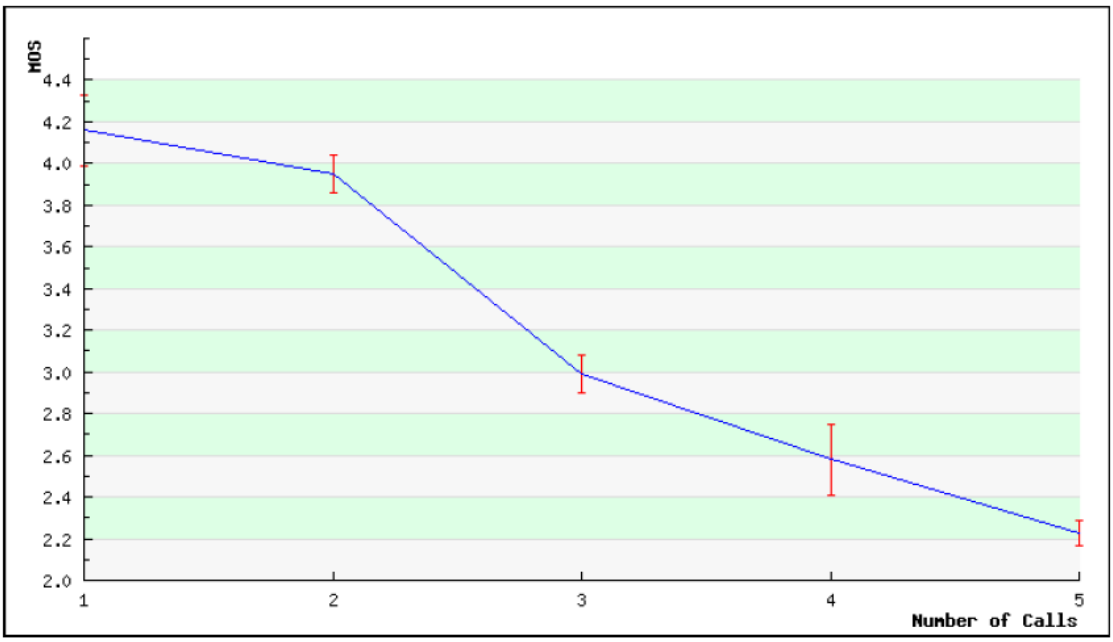


Figura 30 – Política SFQ em Linux com buffer de 16 PKT's

Nestes dois últimos gráficos podemos notar que em uma política SFQ em roteadores Linux , usando diferentes limites de buffer com o mesmo tráfego de fundo, nos mostra que utilizando um limite de buffer menor foi realizada pelo menos uma chamada.

5 - No exemplo abaixo utilizamos um enlace de 100 Mbps e filtro de 8 Mbps, em um roteador Linux

Política FIFO e tráfego de fundo de 8 Mbps

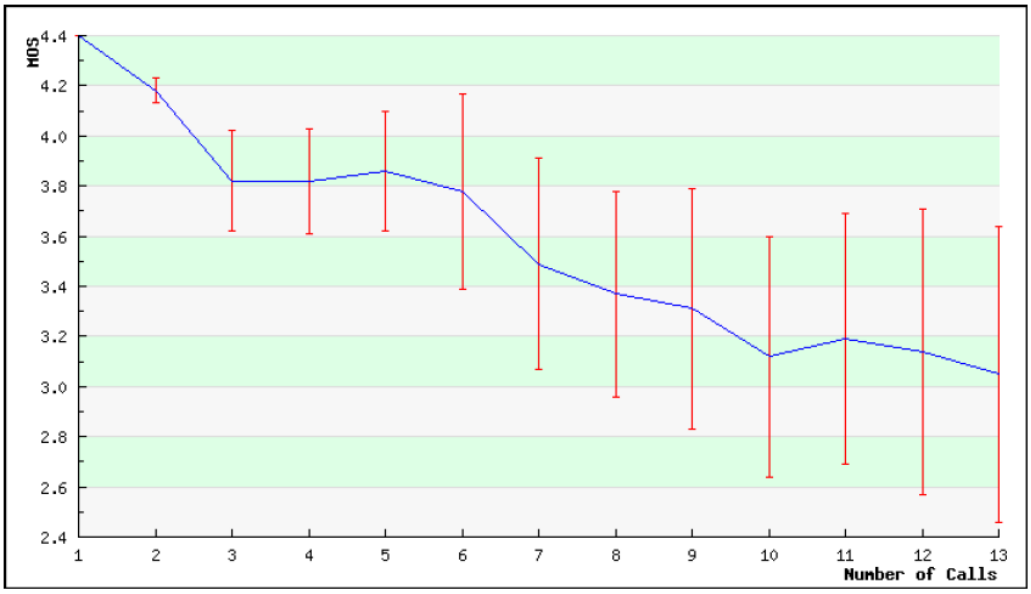


Figura 31 – Política FIFO Linux com tráfego de fundo de 8 Mbps

Política SFQ (16 pacotes de buffer) tráfego de fundo de 8 Mbps

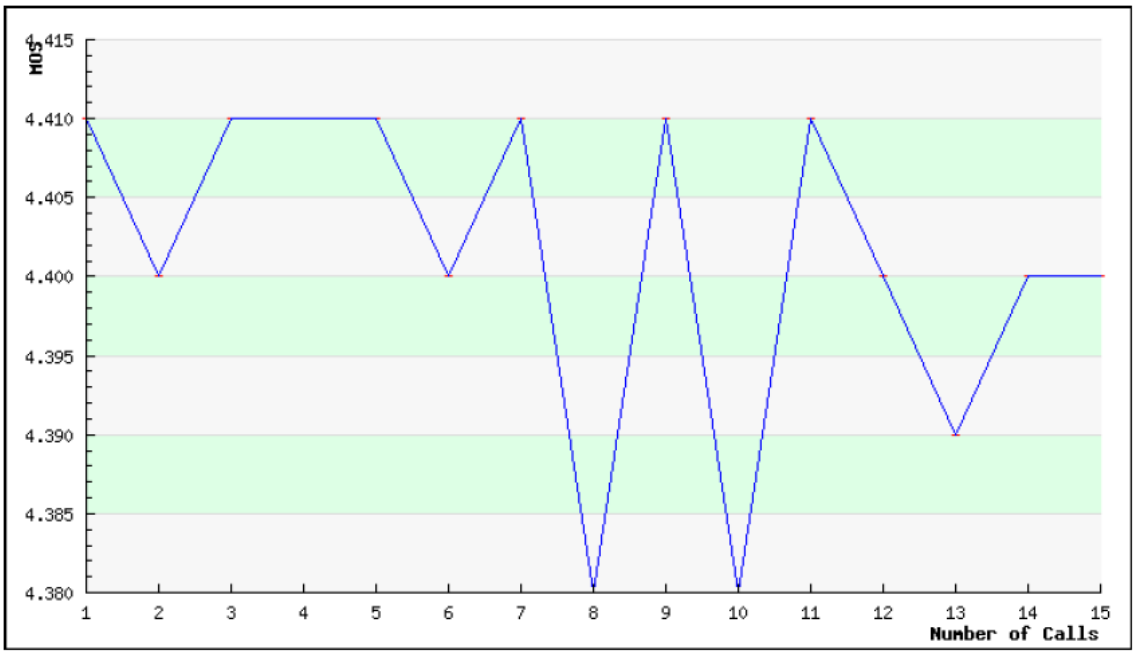


Figura 32 – Política SFQ Linux com buffer de 16 PKT's e tráfego de fundo de 8Mbps

Política PRIO com tráfego de fundo de 8 Mbps

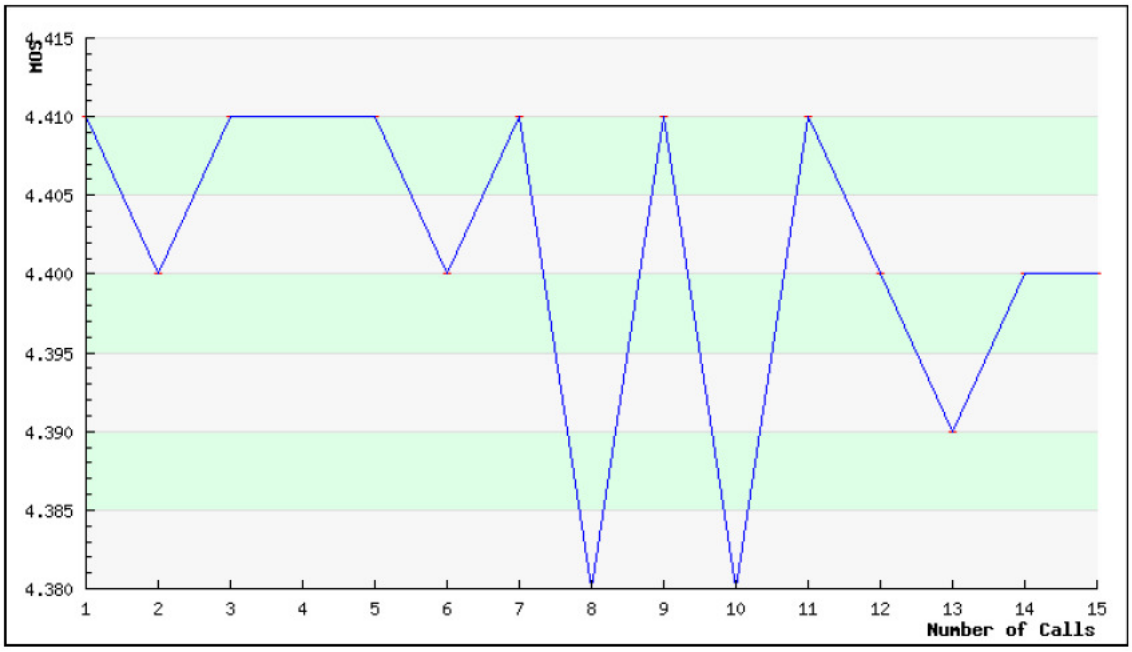


Figura 33 – Política PRIO Linux com tráfego de fundo de 8Mbps

Utilizando políticas de tráfego como SFQ e PRIO em sistemas maiores, isto é, com maior largura de banda é preferível a sistemas de menor largura de banda, pois sistemas menores tende a ser de alta utilização, podemos comparar isto nos gráficos da figura 30 e 32.